*SIMMER-II: A Computer Program for
LMFBR Disrupted Core Analysis*

W. R. Bohl

L. B. Luck

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

This work was supported by the US Nuclear Regulatory Commission, Containment Systems Research Branch, Division of Accident Evaluation, Office of Nuclear Regulatory Research.

An Affirmative Action/Equal Opportunity Employer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

CONTENTS

ABSTRACT	1
I. SIMMER-II COMPUTER PROGRAM OUTLINE	I-1
II. INTRODUCTION	II-1
A. A Brief Summary of SIMMER-II Models and Methods	II-2
B. SIMMER-II Fluid-Dynamics Programming	II-5
C. SIMMER-II Neutronics Programming	II-5
D. SIMMER-II Modeling Philosophy	II-6
E. Summary of the SIMMER-II User's Manual	II-7
REFERENCES	II-7
III. SIMMER-II PHYSICAL MODELS	III-1
A. SIMMER-II Geometry	III-1
B. Fluid-Dynamics and Structural Models	III-1
1. Subscript Notation	III-4
2. Volume Fraction Treatment	III-5
3. The Structure Field	III-14
4. The Liquid Field	III-20
5. The Vapor Field	III-26
C. Equation-of-State Models	III-29
1. Analytic Equation of State	III-29
a. Material Temperatures	III-29
b. Two-Phase Flow Pressures	III-35
c. Single-Phase Flow States	III-38
d. Liquid Microscopic Densities	III-42
2. Tabular Equation-of-State Model	III-42
a. Material Equations of State	III-42
b. Vapor Mixture Rule	III-43
3. Vapor Mixture Properties	III-46
D. Exchange-Function Models	III-49
1. Structure-Configuration Model	III-49
2. Can-Wall Failure Model	III-62
3. Liquid-Droplet Particle-Size Model	III-66
4. Energy-Exchange Models	III-70
a. Structure-Structure Heat Transfer	III-70
b. Liquid-Structure Heat Transfer	III-71
c. Liquid-Liquid Heat Transfer	III-73
d. Structure-Vapor Heat Transfer	III-75
e. Liquid-Vapor Heat Transfer	III-76
5. Momentum-Exchange Models	III-78
a. Momentum Exchange with the Structure Field	III-78
b. Liquid-Field--Vapor-Field Momentum Exchange	III-83
c. Drag Heating	III-85
d. Orifice Pressure Drop	III-86
e. Particle Jamming Restriction	III-88
f. Plug Modeling	III-90
6. Mass-Exchange Models	III-90
a. Solid-Component Breakup Rates	III-90
b. Liquid-Component Freezing Rate	III-93
c. Simple Vaporization-Condensation Model	III-97
d. Multicomponent Phase-Transition Model	III-101
e. Fission Gas Release Model	III-111

E.	Neutronics and Cross-Section Models	III-117
1.	Time-Dependent Neutron Transport Equation	III-117
2.	Time-Dependent Neutron Diffusion Model	III-121
3.	Point-Kinetics Model	III-121
4.	Cross-Section Model	III-123
REFERENCES		III-124
IV.	FLUID-DYNAMICS AND STRUCTURE METHODS	IV-1
A.	Spatial Computing Mesh	IV-4
B.	Mesh-Cell Input	IV-5
C.	Exchange-Function Methods	IV-17
1.	Performing Desired Geometric Modifications	IV-19
2.	Determining Component Temperatures	IV-21
3.	Assigning Control and Fuel Subassemblies to the Same Mesh Cell	IV-24
4.	Structure-Configuration Method	IV-26
5.	Implicit Calculation of the Liquid Droplet Size	IV-31
6.	Implicit Heat-Transfer Calculation	IV-38
7.	Methods for the Simple Vaporization-Condensation Model	IV-46
8.	Solution Method for the Multicomponent, Phase- Transition Model	IV-58
9.	Completion of the Exchange Functions	IV-64
D.	Plug Modeling	IV-66
E.	Explicit Calculation	IV-71
1.	Explicit Continuity Equations	IV-71
2.	Explicit Energy Equations	IV-79
3.	Advanced-Time Volume Fractions	IV-81
4.	Single-Phase-Flow Explicit Calculation	IV-86
5.	Two-Phase-Flow Explicit Calculation	IV-89
6.	Single-Phase/Two-Phase Transition	IV-91
7.	Other Single-Phase-Flow Cases	IV-92
F.	Implicit Calculation	IV-92
1.	Explicit Portions of the Momentum Equations	IV-92
2.	The Velocity Solution of the Momentum Equations	IV-101
3.	Potential Modification of Two-Phase Cells	IV-104
4.	The Implicit Mass-Continuity Equation	IV-105
5.	The Pressure-Change Estimate	IV-106
6.	Final-State Pressure, Density, and Velocity Estimates	IV-107
7.	Pressure Iteration Summary	IV-109
8.	Liquid-Field and Vapor-Field Component Final Densities	IV-109
9.	Vapor Field Final Energy	IV-111
G.	Boundary Conditions and Fluid-Dynamics Controls	IV-112
1.	Rigid, Free-Slip Boundary Condition	IV-113
2.	Boundary Pressure as a Function of Time	IV-114
3.	Rigid, No-Slip Boundary Condition	IV-114
4.	Continuative Outflow (or Inflow) Boundary Condition	IV-114
5.	Fluid-Dynamics Controls	IV-115
a.	Initial Time Step, DTL(1)	IV-115
b.	Long Edit, DTL(2)	IV-115
c.	Courant Condition on Material Velocity, DTL(3)	IV-115
d.	Maximum Vapor Volume Fraction Change, DTL(4)	IV-116
e.	Maximum Time-Step Increase, DTL(5)	IV-118
f.	Maximum Time Step, DTL(6)	IV-118
g.	Optimum Pressure Iteration, DTL(7)	IV-118
h.	Phase-Transition-Rate Limits, DTL(8)	IV-118
i.	High Reactivity Time-Step Limitation, (DTL)9	IV-119

j.	Neutronics Time-Step Control, (DTL)10	IV-120
k.	Vapor Specific Internal Energy Change, DTL(11)	IV-120
l.	Liquid Specific Internal Energy Change, DTL(12)	IV-121
m.	Restart Dumps, DTL(13)	IV-122
n.	Problem Termination	IV-122
H.	Tabular Equation-of-State Interpolation Methods	IV-122
REFERENCES		IV-125
V.	NEUTRONICS CALCULATIVE METHODS	V-1
A.	The Spatial, Temporal, and Energy Mesh	V-1
B.	Methods	V-3
1.	Cross-Section Methods	V-3
a.	Neutronics/Fluid-Dynamics Coupling	V-4
b.	Use of the Neutronics Preprocessor	V-9
c.	Determination of Shielding Factors (NEUPREP-Off Method)	V-15
d.	Determination of Shielding Factors (NEUPREP-On Method)	V-15
e.	Calculation of Macroscopic Cross Sections	V-23
f.	Calculative Approach Used by SHLDXS (NEUPREP-Off Method)	V-23
g.	Calculative Approach Used by SHLDXS (NEUPREP-On Method)	V-23
2.	Quasistatic Method	V-31
a.	General Approach	V-31
b.	Quasistatic Equations	V-34
3.	Discrete-Ordinates Method	V-45
a.	Time-Dependent Discrete-Ordinates Equations	V-45
b.	Fine-Mesh Rebalance Equations	V-48
c.	Boundary Conditions	V-49
d.	Adjoint Stationary Equations	V-50
4.	Diffusion Theory Method	V-51
a.	Time-Dependent Diffusion Equation	V-51
b.	Boundary Conditions	V-52
c.	Adjoint Stationary Equation	V-52
5.	Delayed-Neutron Precursors	V-53
C.	Solution Algorithms	V-54
1.	Flux-Shape Solution	V-54
a.	Inner Iterations for the Discrete-Ordinates Method	V-55
b.	Inner Iterations for the Diffusion Theory Method	V-61
c.	Matrix Solution Techniques for Inner Iteration Solution Methods	V-62
d.	Outer Iterations	V-65
e.	Adjoint Equation Solution	V-71
2.	Quasistatic Calculative Flow	V-72
a.	Time-Step Controls	V-74
b.	Initialization	V-81
c.	Reactivity Evaluation Steps	V-82
d.	Flux-Shape Evaluation Steps	V-87
D.	Point Reactor Model (URANUS) Option	V-87
1.	Point Reactor Model Equations	V-88
2.	Energy Source	V-90
3.	Point Reactor Model Calculative Flow	V-91
4.	Point Reactor Model Time-Step Controls	V-91
REFERENCES		V-92

VI.	PROGRAM DESCRIPTION	VI-1
A.	Overall Program Structure	VI-1
1.	Creating the FORTRAN Compiler Source	VI-1
a.	SIMMER-II Source File	VI-2
b.	Description of the SIMMER-II Code Options	VI-3
2.	Deck Descriptions	VI-7
3.	Program Calculative Flow	VI-11
B.	Programming	VI-31
1.	Fluid-Dynamics Programming	VI-32
2.	Tabular Equation-of-State Programming	VI-35
a.	Structure of the SESAME Library	VI-35
b.	LCM Storage and Pointer Words	VI-35
c.	Calculation of Thermodynamic Functions	VI-39
3.	Space-Dependent Neutronics	VI-41
a.	Small-Core Storage	VI-41
b.	Large-Core Storage	VI-42
4.	Cross-Section Storage and Programming	VI-42
a.	Programming the NEUPREP-Off Version of SHLDXS	VI-47
b.	Programming the NEUPREP-On Version of SHLDXS	VI-51
5.	Data Common to Neutronics and Fluid-Dynamics Calculations	VI-54
C.	Data Mangement	VI-54
1.	Standard Files	VI-54
2.	Internal Files	VI-55
D.	Input Description	VI-59
E.	Printed Output Description	VI-59
1.	Problem Control Card Input Edit	VI-60
2.	Space-Dependent Neutronics Input Edit	VI-60
3.	Fluid-Dynamics Input Edit	VI-61
4.	Neutronics Initial Calculation Edit	VI-62
5.	Neutronics Transient Calculation Edit	VI-64
6.	Fluid-Dynamics Transient Calculation Edit	VI-66
a.	Fluid-Dynamics Time-Step Edits	VI-68
b.	Fluid-Dynamics Long Prints	VI-73
c.	Short Print Summations	VI-73
d.	Diagnostic Information and Guidance on Spurious Results	VI-73
F.	Plotted Output	VI-77
G.	Restart Files	VI-83
1.	Restart File Structure	VI-83
2.	Restart File Writing and Reading	VI-84
H.	Postprocessing Features	VI-85
I.	Computing-System-Dependent Features	VI-85
1.	LTSS FTNLTS System	VI-85
2.	CDC-7600 SCOPE System	VI-86
3.	IBM Operating System	VI-86
4.	Digital Equipment Corporation (DEC) Operating Systems	VI-87
5.	System-Dependent Routines	VI-87
	REFERENCES	VI-87
	ACKNOWLEDGMENTS	VI-88
	APPENDIX A: NOMENCLATURE	A-1
	APPENDIX B: SINGLE-PHASE EQUATION OF STATE	B-1

APPENDIX C:	COALESCENCE MODEL	C-1
APPENDIX D:	LIQUID-LIQUID HEAT-TRANSFER MODEL	D-1
APPENDIX E:	SPECIAL CASES IN THE PHASE-TRANSITION MODEL	E-1
APPENDIX F:	THE PRESSURE ITERATION MATRIX	F-1
APPENDIX G:	MATRIX INVERSION METHODS	G-1
APPENDIX H:	TRANSIENT INITIAL STATE CALCULATION FOR THE SPATIAL NEUTRONICS	H-1
APPENDIX I:	DERIVATION AND CALCULATIVE APPROACH FOR THE MULTIGROUP NEUTRON DIFFUSION EQUATIONS	I-1
APPENDIX J:	SIMMER-II COMMON VARIABLE DICTIONARY	J-1
APPENDIX K:	SIMMER-II CONSTANTS	K-1
APPENDIX L:	DERIVATION AND CALCULATIVE APPROACH FOR THE MULTIGROUP NEUTRON TRANSPORT EQUATIONS	L-1
APPENDIX M:	CALCULATIVE APPROACH AND TREATMENT OF THE QUASI-STEADY AMPLITUDE EQUATION	M-1
APPENDIX N:	SIMMER-II SUBROUTINES, FUNCTIONS, AND COMMON DECK	N-1
APPENDIX O:	AASC ARRAY EQUIVALENCES	O-1
APPENDIX P:	DESCRIPTION OF NEUTRONICS COMMON ARRAYS	P-1
APPENDIX Q:	SIMMER-II INPUT DATA	Q-1
APPENDIX R:	CONSIDERATIONS ON ERRORS IN SHIELDING FACTOR CALCULATION RESULTING FROM COMMON ISOTOPES IN MATERIAL CROSS SECTIONS . .	R-1
APPENDIX S:	SYSTEM-DEPENDENT FUNCTIONS AND SUBROUTINES	S-1

FIGURES

1.	SIMMER-II cylindrical geometry	III-2
2.	SIMMER-II mass-transfer paths	III-15
3.	Typical pressure isotherm with Van der Waals loop	III-44
4.	Boundary-layer energy exchange during phase transition	III-102
5.	Boundary-layer mass exchange during phase transition	III-102
6.	Neutron transport coordinates in (r,z) geometry	III-119
7.	Coupled neutronics-fluid dynamics calculative flow	IV-2
8.	SIMMER-II fluid-dynamics computing mesh	IV-5
9.	SIMMER-II fluid-dynamics mesh cell	IV-6
10.	Mesh point sets	IV-7
11.	Structure-configuration model flow diagram	IV-27
12.	Node structure for defining the initial plug position	IV-67
13.	Plug interface tracking scheme	IV-68
14.	SIMMER-II treatment of the fluid dynamics at the plug interface	IV-70
15(a).	Overall structure of the cross-section shielding and mixing calculations (NEUPREP-off option)	V-26
15(b).	Structure of the background and temperature interpolation portion of the cross-section shielding-factor calculation (NUEPREP-off option)	V-27
16.	Structure of cross-section shielding and mixing calculations (NEUPREP-on option)	V-28
17.	Structure of the background interpolation portion of the shielding-factor calculations (NEUPREP-on option)	V-29
18.	Structure of the temperature interpolation portion of the shielding-factor calculations (NEUPREP-on option)	V-30
19.	General approach used in the overall quasistatic algorithm calculative flow	V-73
20.	Functions used for time-step control of the quasistatic method algorithm	V-76
21.	Calculative flow of the quasistatic method algorithm for the extrapolated flux-shape option	V-84
22.	Calculative flow of the quasistatic method algorithm for the updated flux-shape option	V-85
23.	Calculative flow of the 7-iteration portion of the quasistatic method algorithm for both flux-shape options	V-86
24.	SIMMER-II overlay structure	VI-8
25.	SIMMER-II calculative flow	VI-12
26.	Overall input calculative flow	VI-13
27.	Fluid-dynamics input calculative flow	VI-15
28.	Transient calculative flow	VI-16
29.	Neutronics calculative flow	VI-17
30.	Subroutine OUTER calculative flow	VI-20
31.	Neutron diffusion calculative flow in subroutine INNED	VI-23
32.	Neutron transport calculative flow in subroutine INNED	VI-24
33.	Quasistatic method calculative flow	VI-26
34.	Fluid-dynamics calculative flow	VI-27
35.	AEOS calculative flow	VI-30
B-1.	Pressure-volume relation for the single-phase EOS	B-3
C-1.	Physical concept for the droplet-droplet collision frequency	C-2
D-1.	Physical representation of liquid-liquid interaction	D-2
D-2.	Collision configuration for liquid m and liquid k	D-4
D-3.	Assumed contact interval	D-6

D-4.	Heat-transfer treatment during contact	D-8
E-1.	Plot of F as a function of T_{sat} from Eq. (IV-100) for a typical situation with convergence difficulties	E-5
E-2.	Enlargement of the F function in the vicinity of the critical temperature	E-5
L-1.	The ordering of the S_4 directions	L-7
L-2.	The ordering of the S_8 directions	L-8
M-1.	Calculative flow for the amplitude equation solution	M-9

TABLES

I.	DENSITY AND ENERGY COMPONENT SUBSCRIPTS	III-6
II.	DENSITY EQUIVALENCES FOR ENERGY COMPONENTS	III-7
III.	COMPONENT EQUATION-OF-STATE ASSIGNMENTS	III-8
IV.	STRUCTURE-FIELD MASS-TRANSFER RATES	III-17
V.	LIQUID-FIELD MASS-TRANSFER RATES	III-21
VI.	SIMMER-II REGION PARAMETERS	III-52
VII.	FAIL3 VALUES FOR CAN-WALL FAILURE	III-65
VIII.	EXCHANGE-FUNCTION CALCULATIVE SEQUENCE	IV-18
IX.	STRUCTURE-FIELD ENERGY COMPONENT INDICATORS	IV-31
X.	MESH-CELL TYPE	IV-32
XI.	STRUCTURE-FIELD SURFACE TEMPERATURE INDEX VALUES	IV-40
XII.	STORAGE VARIABLES FOR MASS-TRANSFER RATES	IV-74
XIII.	DEFINITION OF THE STRUCTURE-FIELD DENSITY COMPONENT INDIVIDUAL MASS-TRANSFER RATES	IV-76
XIV.	DEFINITION OF THE LIQUID-FIELD DENSITY COMPONENT INDIVIDUAL MASS-TRANSFER RATES	IV-77
XV.	DEFINITION OF THE VAPOR-FIELD DENSITY COMPONENT INDIVIDUAL AND TOTAL MASS-TRANSFER RATES	IV-78
XVI.	DECISION MATRIX FOR THE TWO-PHASE/SINGLE-PHASE TRANSITION IN EXFLUD.	IV-85
XVII.	UNSHIELDED MICROSCOPIC CROSS SECTIONS USED BY SIMMER-II	V-6
XVIII.	ISOTOPIC SHIELDING FACTORS	V-7
XIX.	INTERPOLATION FORMULAS CHOSEN FOR THE BACKGROUND CROSS SECTION	V-18
XX.	TEMPERATURE INTERPOLATION FORMULAS	V-21
XXI.	MACROSCOPIC CROSS SECTIONS	V-24
XXII.	SUBROUTINE GRIND CALCULATIVE FLOW FLAGS	VI-19
XXIII.	DESCRIPTION OF SESAME DATA RECORDS	VI-36
XXIV.	DIRECTORY FILE FOR THE SESAME LIBRARY	VI-37
XXV.	INDEX RECORD FOR A SESAME MATERIAL FILE	VI-38
XXVI.	INTERNAL VARIABLES USED IN ROUTINE P4	VI-40
XXVII.	MICROSCOPIC CROSS-SECTION MIXING INSTRUCTIONS	VI-45
XXVIII.	SCM STORAGE POINTERS FOR UNSHIELDED MICROSCOPIC CROSS SECTIONS AND SHIELDING FACTORS	VI-50
XXIX.	PLOTTED FLUID-DYNAMICS MESH-CELL VARIABLES	VI-79
E-I.	INDICES USED FOR THE INNER ITERATION ON THE CONSERVATION EQUATIONS AND THE SATURATION TEMPERATURE	E-7
I-I.	DEFINITION AND LOCATION OF CALCULATIVE VARIABLES USED FOR DIFFUSION THEORY	I-14
J-I.	SIMMER-II COMMON BLOCK FORTRAN VARIABLES	J-1
K-I.	SIMMER-II CONSTANTS	K-1
L-I.	DEFINITION AND LOCATION OF CALCULATIVE VARIABLES USED IN THE TRANSPORT EQUATIONS	L-26
N-I.	SUBROUTINE FUNCTIONS	N-1
N-II.	SIMMER-II COMMON DECK FUNCTIONS	N-11
N-III.	OVERLAY ENTRY POINTS	N-13
O-I.	FORTRAN VARIABLES EQUIVALENCED TO THE AASC CELL STORAGE BLOCK	O-1
P-I.	CONTENTS OF THE IA COMMON ARRAY	P-1
Q-I.	OPTIONS FOR SPECIAL READ FORMATS	Q-12
R-I.	TYPICAL MIXING ERROR IN THE CAPTURE REACTION RESULTING FROM ²³⁸ U AS A COMMON ISOTOPE	R-5

R-II.	TYPICAL MIXING ERROR IN THE CAPTURE REACTION RESULTING FROM ^{16}O AS A COMMON ISOTOPE	R-6
R-III.	ILLUSTRATION OF THE DOMINANT ISOTOPE SEPARATION TREATMENT	R-8
R-IV.	AN EXAMPLE OF THE ACCURACY IMPROVEMENT POSSIBLE FOR THE MIXING METHOD USING THE COMMON ISOTOPE SEPARATION TREATMENT	R-9

SIMMER-II: A COMPUTER PROGRAM FOR LMFBR DISRUPTED CORE ANALYSIS

by
W. R. Bohl and L. B. Luck

Primary Contributors

F. R. Parker
C. R. Bell
L. L. Smith

ABSTRACT

SIMMER-II (Version 12) is a computer program to predict the coupled neutronic and fluid-dynamics behavior of liquid-metal fast reactors during core-disruptive accident transients. The modeling philosophy is based on the use of general, but approximate, physics to represent interactions of accident phenomena and regimes rather than a detailed representation of specialized situations.

Reactor neutronic behavior is predicted by solving space (r,z), energy, and time-dependent neutron conservation equations (discrete ordinates transport or diffusion). The neutronics and the fluid dynamics are coupled via temperature- and background-dependent cross sections and the reactor power distribution. The fluid-dynamics calculation solves multicomponent, multiphase, multifield equations for mass, momentum, and energy conservation in (r,z) or (x,y) geometry. A structure field with nine density and five energy components; a liquid field with eight density and six energy components; and a vapor field with six density and one energy component are coupled by exchange functions representing a modified-dispersed flow regime with a zero-dimensional intra-cell structure model.

I. SIMMER-II COMPUTER PROGRAM OUTLINE

1. Program Identification: SIMMER-II
2. Computer for Which Program Is Designed: Originally the CDC-7600. However, calculations at the Los Alamos National Laboratory are performed almost exclusively on Cray computers. The code also has been run on various versions of VAX-11 and IBM computers.
3. Description or Function: SIMMER-II is designed to solve coupled space- and time-dependent neutronics and multiphase, multicomponent, Eulerian fluid dynamics equations to predict the two-dimensional (r,z) or (x,z) motion of LMFBR core materials during a core-disruptive accident.
4. Method of Solution: The neutronics calculation in SIMMER-II uses a quasistatic method combined with standard finite-difference techniques to solve either multigroup diffusion or discrete-ordinates transport equations in two spatial dimensions and time. The point kinetics equations alternatively may be solved if desired. The fluid-dynamics calculation solves multicomponent, multifield coupled equations for mass, momentum, and energy conservation. Nine density and five energy components are used in the structure field; eight density and six energy components are used in the liquid field; six density components and one energy component are used in the vapor field. A combination of explicit and implicit time differencing and donor-cell spatial finite differencing techniques is used. Both an analytic and a tabular equation of state are provided as options.
5. Restrictions: For the neutronics, anisotropic scattering is treated approximately. Although the cell-wise cross sections depend on temperature and self-shielding, other fine-structure spectral effects are not included. Only (r,z) cylindrical geometry is allowed for a neutronics calculation, and no provision is made for transport of delayed-neutron precursors. The fluid-dynamics models are restricted to the components described in Item 4 above. Only the flow regime of dispersed droplets is used to calculate the interfacial phenomena between the components. A simple zero-dimensional fuel pin model is used to represent the within-cell structure where appropriate. Mechanical deformation of structure is not modeled. Current dimension statements limit a problem to a total of 1000 mesh cells with a limit of 100 cells in either the axial or horizontal direction.

6. **Running Time:** It takes less than 1 min to compile and load SIMMER-II on a Cray-1 computer. The run time depends on the options selected and the problem size. Approximately 37 000 lines of code are compiled.
7. **Unusual Features of the Program:** The program is organized to allow various models to be selected using options available in the code and accessible through a commercial code management system provided by the user.* Storage is divided between small-core memory (SCM) and large-core memory (LCM) for CDC computers. For Cray computers, this scheme is simulated for the neutronics portion of the code and eliminated in the fluid dynamics portion. The output can be on microfilm, microfiche, and/or printer, depending on the packages available at a particular installation. A flexible postprocessor is provided for more extensive output analyses, including options for movie generation. An optional neutronics preprocessor is provided to improve neutronics calculative efficiency and physical modeling.
8. **Machine Requirements:** At most six files may be needed for input including data, cross sections, equation of state, and dump restarts. Depending on the options selected, six files are used for output and five files are used for data manipulation. The minimum SCM needed is 65 000 words. The minimum LCM needed is 50 000 words, whereas up to 300 000 words of LCM may be necessary depending on the problem size. A variable memory option is provided to limit LCM requirements to the space actually used.
9. **Related Programs:** KACHINA, TWOTRAN, SIMMER-I.
10. **Material Available:** Source deck, test problems, results of executed test problems, and this report.

*The management system should be equivalent to CDC's UPDATE or OP CODE's HISTORIAN system.

II. INTRODUCTION

The SIMMER-II (S_n, Implicit, Multifield, Multicomponent, Eulerian, Recriticality) computer program was released in October 1978.¹ It was developed to alleviate several deficiencies in the SIMMER-I program,² which already had been used to perform accident sequence and consequence calculations and had provided new insight into several important areas for liquid-metal-cooled, fast-breeder reactor (LMFBR) safety. The SIMMER-II code resulted from recognition of the importance of (1) performing detailed calculations for the phenomena that occurred during the transition phase³ and (2) using experimentally supported correlations dependent on local conditions for mass, momentum, and energy transfer occurring in fast-reactor accidents. In the SIMMER-II code, the three SIMMER-I components (sodium, steel, and fuel) have been increased to track intermixing fuel from different enrichment zones and to include the effects of fission gas and control material. Furthermore, the modeling of mass, momentum, and energy transfer incorporates correlations based on experimental results and has an explicit dependence on local conditions (that is, geometry, field velocities, and material properties).

Even as SIMMER-II was being documented and released, method and model improvements were being incorporated into it. Thus, as with many computer programs, the program documentation did not describe the current version of the code adequately. As a result, a revised manual was issued in 1980 that reflected the SIMMER-II code used for studies presented in the 1979 Seattle Fast Reactor Safety conference.⁴ Since the revised manual was issued, some large-scale SIMMER-II studies have been performed.^{5,6} These have resulted in additional code modifications and model changes, and therefore a second revision of the SIMMER-II manual is appropriate.

The new features incorporated into SIMMER-II have resulted from a consideration of the relative need for each new or improved model as well as the time required to develop and incorporate the model. Thus, the addition of more fields, dimensions, and flow regimes has been postponed so that more components and detail could be included in the treatment of transport processes. More fields and dimensions require lengthy model and method development. A multiple flow-regime treatment could be implemented into the SIMMER-II framework relatively easily, but an extensive review of appropriate and consistent flow-regime correlations is necessary before incorporating such models in the code. Also, the stability of the current fluid dynamics numerical algorithm is questionable

in an arbitrary flow-regime environment. These and other considerations and tradeoffs should be evaluated before developing model and method improvements.

A. A Brief Summary of SIMMER-II Models and Methods

SIMMER-II couples space-dependent neutronics with multifield, multicomponent fluid dynamics in a cylindrical coordinate (r,z) geometric representation. The code includes different neutronics models to allow the user trade-offs in accuracy and efficiency. A time-dependent neutron-transport option is included to calculate accurately the neutronics effects of large-scale material motion during a transient. When the underlying assumptions are applicable or the greater calculative efficiency is required, a neutron diffusion option can be used to include space-dependent effects without solving the transport equation. The quasistatic method⁷⁻⁹ is used to solve for the time-dependence of the flux in both the transport and diffusion options. Cell-wise cross sections in the space-dependent neutronics calculation are determined at each neutronics time step using the Bondarenko formalism.¹⁰ Using a neutronics preprocessor before an accident calculation enables a SIMMER-II option that results in an improved physics treatment and significant computing efficiency gains. Additional options are included for performing point kinetics neutronics or using input values of power vs time or of excess reactivity vs time for special-purpose studies of whole-core accidents or small-scale studies of single subassembly phenomena.

The SIMMER-II fluid-dynamics model has evolved from and still bears some resemblance to the KACHINA¹¹ program, in which the relative motion of liquid and vapor fields was calculated using the Implicit Multifield (IMF) method. The KACHINA model was extended in SIMMER-I and again in SIMMER-II by adding density and energy components in each field to allow the motion of the various LMFBR materials. A structure field also was added to model solid fuel, cladding, and subassembly walls (also referred to as subassembly can-walls or can-walls). As in KACHINA, within each field all materials or components move with the velocity of the field. This means that all the liquid field components have the same velocity at each point in space. However, multiple energy equations are solved in each field except for the vapor field where the mixture of vapors is treated with a single energy equation. The number of energy equations does not equal the number of continuity equations; for example, SIMMER-II tracks the macroscopic density for both fertile and fissile fuel components. These two components currently can represent any number of fuel enrichments and burnups without having a fuel component for each enrichment, provided the isotopic composition of a

component is constant over time. For example, the fertile and fissile components would be uranium dioxide and plutonium dioxide, respectively, in a mixed-oxide fueled LMFBR. However, the two components are assumed to be mixed such that they have a common temperature, and only a single energy is needed to describe the thermodynamic state of the two components.

The structure field is fixed in space and acts as an infinite momentum sink. In the structure field, a distinction is made between fabricated fuel pellets and fuel that, once molten, freezes onto other structures. The primary reason for this distinction is that fuel crust formation is very important in determining energy transfer to can-walls. Such crusts now are tracked separately from fuel pellets, which may co-exist at the same axial location. A further increase in the number of the energy components results from allowing solid particles to exist in the liquid field. In SIMMER-II, these particles are composed either of fuel or steel and result from molten fuel and steel freezing without adhering to a structure or are a consequence of structural fuel and steel disintegrating before becoming completely molten. Assigning a high viscosity to these particles, allows them to simulate the limited penetration of fuel beyond the core periphery, which might occur initially in a LMFBR meltdown transient.

Fission gas and control material are included in the SIMMER-II components. Control material has been introduced primarily to determine its neutronics effects but could provide some understanding as to its structural and fluid-dynamics role in accident analysis. Therefore, control components have been included in all three fields, but the fields do not account for frozen control material and control-material solid particles separately. Accident investigations with SIMMER-II and the obtaining of additional knowledge of the B_4C equation of state will help to determine the need for future model developments for predicting control-material behavior. Both intragranular (within fuel grains) and intergranular (on grain boundaries between fuel grains) fission-gas components are included in the SIMMER-II program. These two components serve as a mechanism for predicting the fission-gas release rate to the vapor field. In the vapor field, not only does the fission-gas component contribute to the pressure (in which case it may be a dispersive mechanism), but it also can impede the condensation of the fuel, steel, sodium, and control-vapor components as a result of inter-cell transport and accumulation of fission gas.

The improved treatment of mass, momentum, and energy-transfer functions in SIMMER-II is as important as the addition of more components. The exchange functions are designed primarily to treat phenomena associated with LMFBR acci-

dent analysis; however, significant flexibility is provided to examine model sensitivities and to provide a means for experiment calibration.

A major aspect of the improved exchange-function treatment is the structure-configuration model. The model first determines the geometric arrangement of the structure-field energy components. Although this arrangement depends on the information the user provides, the fuel pins and the subassembly can-wall generally provide the basis for the arrangement of the components. The model then determines the order in which the frozen solid materials are layered on the pin and can-wall structures. With this, the model adjusts the flow channel hydraulic diameter for the loss or gain of solid material in the structure field from melting or freezing. The model next computes the heat-flow paths, layer thicknesses, heat-transfer coefficients, and contact surface areas for the layered components. As a consequence, the structure-configuration model defines the exterior structure surfaces seen by the flow fields, including the surface heat-transfer coefficients and surface areas. Special options are included to model fuel motion in intersubassembly gaps and to set up coefficients for heat transfer to can-walls in adjoining cells after a given cell's can-wall has melted.

The flow-regime treatment of SIMMER-II currently is the same as in KACHINA; that is, the liquid field is treated as dispersed droplets except when low vapor-volume fractions occur. In the latter case, a bubbly flow-regime is used for phase transitions. In SIMMER-II, droplet radii for the dispersed-flow regime are computed dynamically and reflect physical phenomena occurring locally in space and time. As a result of the structural configuration and improved dispersed-flow models, both heat-transfer coefficients and drag functions are calculated using common engineering correlations for interactions between the fields and their energy components.

The basic vaporization-condensation model in SIMMER-II is the same as the conduction-limited model in SIMMER-I. However, the SIMMER-II solution technique is more complex, requiring an iterative procedure to obtain a more implicit solution. A more accurate (but still developmental) multicomponent vaporization-condensation model that includes the effects of noncondensables and multicomponent interference and calculates the condensation of one vapor material on a different liquid material (for example, fuel vapor on liquid sodium) has been added.

A nonequilibrium model is used to calculate melting or freezing in SIMMER-II. This means crusts can be formed on cold surfaces from fuel whose bulk temperature is above its melting temperature; if the liquid is hot enough, steel below the melting temperature can be ablated into the liquid field. In addition, the inclusion of solid particles in the liquid field permits the transfer of solid material from the structure to the liquid field.

The analytic equation of state used in SIMMER-II includes all five materials treated in SIMMER-II: fuel, steel, sodium, control, and fission gas. A physically consistent transition between single- and two-phase flow is provided. In addition, the gas "constants" depend on the partial pressures of the individual components, and the liquid microscopic densities are temperature dependent for fuel, steel, and sodium.

B. SIMMER-II Fluid Dynamics Programming

The SIMMER-II fluid-dynamics models have been reprogrammed from that in previous versions. The main fluid-dynamics effort is separated into three primary code functions: an exchange function calculation, an explicit calculation, and an implicit calculation. Within these, the number of subroutines has grown as the SIMMER-II models have been extended. The SIMMER-II code also has been modified so that versions can be compiled that are suitable for use on Cray computers. The two-level memory structure originally imposed by the CDC architecture has been retained in the neutronics part of the code but not in the fluid-dynamics part. The disc-based shuffle of data between the neutronics and fluid dynamics calculations has been made an option, and the mnemonics used for variable names has been changed for consistency with the mathematical symbols used in this report. Finally, the FORTRAN coding for all the fluid dynamics has been restructured in a minor way and extensively "commented" by our Kernforschungszentrum, Karlsruhe (KfK), partners to make the programming more understandable. This excellent work is contained in the current version of SIMMER-II and is described in KfK reports.^{12,13,14}

C. SIMMER-II Neutronics Programming

Two parts of the neutronics methods have been changed extensively from that described in the previous SIMMER-II manuals. First, the implementation of the quasistatic method for solution of the time dependence of the flux-shape (diffusion or transport) equation was found to be inadequate to treat the strongly coupled neutronics and fluid dynamics of heterogeneous core reactors.^{5,6} The

quasistatic method was not a problem because it is a mathematically exact method for solving the time dependent neutron conservation equations, but numerical approximations used in its implementation were improved by the techniques described in Chap. V. Second, the calculation of cell-wise macroscopic cross sections was found to be excessively time consuming and for some applications was insufficiently accurate. These problems were resolved by the use of premixed material cross sections, precalculated shielding factor representations, and vectorization. A neutronics preprocessor¹⁵ was developed for SIMMER-II users desiring to use this option.

D. SIMMER-II Modeling Philosophy

SIMMER-II has been used in LMFBR safety assessments in the US and other countries and for related applications such as experiment analysis. These applications often have resulted in criticisms of the completeness and sophistication of the SIMMER-II models and methods. Although much of the criticism has been constructive and has enabled us to improve SIMMER-II, it should be recognized that the state of the art is such that a code developer is required to accept approximations and omissions in his treatment of very complex accident phenomena. The developers of SIMMER-II approached their task with the following philosophy and perspective about model and method development and their relationships to reactor accident analysis.

From the beginning of the SIMMER program, it was realized that an accident analysis code could not be built incorporating all of the detailed knowledge available about individual phenomena. Even if a way could be found to connect all the interactions among the various phenomena, no computer yet built could handle the computing task. Further there were, and still are, phenomena about which very little is known. The fundamental problem was to decide which phenomena should be modeled and at what level of sophistication.

Some basic decisions were made. First, SIMMER-II was to be what we have come to call a "system code." That is, modeling detail and accuracy were prudently sacrificed for the ability to integrate models, for example, neutronics and fluid dynamics, and to achieve a reasonable degree of computational economy. As a consequence, SIMMER-II was not designed complete with specialized codes in the analysis of a limited range of phenomena. Second, precise descriptions of accident phenomena were not attempted, rather the physics underlying the desired phenomena were modeled approximately. As a result, SIMMER-II does not have a clad- or a fuel-motion model, and so forth, but it has a heat transfer and fluid

flow model that includes sufficient detail to perform parametric studies on clad and fuel motion. An extra benefit of this approach is that SIMMER-II has a generality not originally anticipated. For example, SIMMER-II has been used successfully for experiment analysis¹⁶ (for example, TREAT tests) and reactor accident applications¹⁷ outside its intended purpose of transition-phase analysis (for example, post-disassembly expansion dynamics). Third, SIMMER-II was not used to obtain a deterministic accident path. This means that, although SIMMER-II could provide information about the effect of different phenomena on the accident progression and about the interaction of different phenomena, any safety assessment for a particular system must involve technical judgments weighing information from all sources--SIMMER-II calculations, special-purpose code calculations, experiment results, and uncertainty analyses. Such a systematic approach evolved from our efforts to analyze accident paths for two different reactor systems, which are discussed in detail in Ref. 18.

E. Summary of the SIMMER-II User's Manual

Chapter III presents a detailed description of the fluid-dynamics and neutronics models used in SIMMER-II. The methods to solve the model equations are in subsequent chapters for the fluid-dynamics and neutronics methods, respectively. A description of the SIMMER-II program design and use is contained in Chap. VI. To make the manual easier to read, much of the detailed material has been placed in appendices.

REFERENCES

1. L. L. Smith, Compiler, "SIMMER-II: A Computer Program for LMFBR Disrupted Core Analysis," Los Alamos National Laboratory report LA-7515-M NUREG/CR-0453 (October 1978), Rev. 1980.
2. C. R. Bell, P. B. Bleiweis, J. E. Boudreau, F. R. Parker, and L. L. Smith, "SIMMER-I: An S_n Implicit, Multifield, Multicomponent, Eulerian, Recriticality Code for LMFBR Disrupted Core Analysis," Los Alamos Scientific Laboratory report LA-NUREG-6467-MS (January 1977).
3. J. F. Jackson, M. G. Stevenson, J. F. Marchaterre, R. H. Sevy, R. Avery, and K. O. Ott, "Trends in LMFBR Hypothetical-Accident Analysis," in "Proceedings of the Fast Reactor Safety Meeting," AEC report CONF-740401 (1979).
4. Proceedings of the International Meeting on Fast Reactor Safety Technology, American Nuclear Society, La Grange Park, Illinois (1979).

5. L. B. Luck, G. P. DeVault, M. W. Asprey, and C. R. Bell, "An Evaluation of the Calculated Results of an Unprotected Transient Undercooling Accident in a Large, Heterogeneous-Core, Liquid-Metal-Cooled Fast Breeder Reactor," Los Alamos National Laboratory report LA-9553-MS, NUREG/CR-3031 (December 1982).
6. T. G. Theofanous and C. R. Bell, "An Assessment of CRBR Core Disruptive Accident Energetics," Los Alamos National Laboratory report LA-9716-MS, NUREG/CR-3224 (March 1984).
7. D. A. Meneley, K. O. Ott, and E. S. Weiner, "Fast-Reactor Kinetics--The QX1 Code," Argonne National Laboratory report ANL-7769 (March 1971).
8. K. O. Ott and D. A. Meneley, "Accuracy of the Quasistatic Treatment of Spatial Reactor Kinetics," Nucl. Sci. Eng. 36, 402-411 (1969).
9. D. A. Meneley, G. K. Leaf, A. J. Lineman, T. A. Daly, and W. T. Sha, "A Kinetics Model for Fast Reactor Analysis in Two Dimensions," in Dynamics in Nuclear Systems (The University of Arizona Press, Tucson, Arizona, 1972).
10. L. P. Abagyan, N. O. Bazazyants, I. I. Bondarenko, and M. N. Nikolaev, Group Constants for Nuclear Reactor Calculations (Consultants Bureau Enterprises, Inc., New York, 1964).
11. A. A. Amsden and F. H. Harlow, "KACHINA: An Eulerian Computer Program for Multifield Fluid Flows," Los Alamos Scientific Laboratory report LA-5680 (December 1974).
12. S. Kleinheins, "SIMMER-II.9 Fluid Dynamics and Point Kinetics Program Structuring, Flowcharts, Tables and Equations," Kernforschungszentrum, Karlsruhe report Primärbericht 01.02.06P01A (August 1984).
13. S. Kleinheins, "SIMMER-II.9K Reorganized Control, Input/Output and Initialization Flowcharts, Tables and Equations," Kernforschungszentrum, Karlsruhe report in Primärbericht 01.02.06P96A (June 1986).
14. S. Kleinheins, "SIMMER-II.10 Flowchart, Table and Equation Updates from SIMMER-II.9," Kernforschungszentrum, Karlsruhe report Primärbericht 01.02.06P66A (March 1985).
15. F. Parker, M. Ishikawa, and L. Luck, "The MXS Cross-Section Preprocessor User's Manual," Los Alamos National Laboratory report LA-10856-M, NUREG/CR-4765 (March 1987).
16. W. R. Bohl, "A SIMMER-II Analysis of the R-7 TREAT Test," Los Alamos Scientific Laboratory report LA-7763-MS, NUREG/CR-0760 (April 1979).
17. C. R. Bell, R. D. Burns III, and L. B. Luck, "Impact of SIMMER-II Model Uncertainties on Predicted Post Disassembly Dynamics," Los Alamos Scientific Laboratory report LA-8053-MS, NUREG/CR-1058 (October 1979).
18. L. B. Luck and C. R. Bell, "The Case for Integral Core-Disruptive Accident Analysis," in "Proceedings of the International Topical Meeting on Fast Reactor Safety," DOE report CONF-850410 (1985).

III. SIMMER-II PHYSICAL MODELS

In this chapter, we describe the physical models used by SIMMER-II to predict the phenomena occurring in an LMFBR core-disruptive accident. This description includes both the model equations and the assumptions used to obtain the models. The model assumptions always should be considered when running SIMMER-II. If a model is extended beyond its range of applicability, the results should be interpreted with caution and the awareness that they may not be valid. Additional assumptions also may be made in implementing the numerical method for a model, and the effect of these assumptions also should be considered.

A. SIMMER-II Geometry

The SIMMER-II geometry is restricted to two dimensions. A right circular cylinder with azimuthal symmetry, as illustrated in Fig. 1, is used for problems in which the fluid-dynamics and neutronics models are coupled. The fluid-dynamics model can be used alone both in this geometry and in two-dimensional (x,z) plane geometry (infinite in the third dimension). Fluid-dynamics analyses can be performed in one dimension by eliminating orthogonal-dimensional dependence. When coupled-neutronics, fluid-dynamics problems are analyzed, the neutronics calculative domain is a subset of the fluid-dynamics calculative domain. Furthermore, the fluid-dynamics mesh spacing can be subdivided for the neutronics calculation. Infinitely dilute cross-sections for the neutronics and certain fluid-dynamics parameters common to a set of mesh cells are assumed to be constant. Fluid-dynamics initial conditions can be specified by region and/or for specific mesh cells. The geometric scope of SIMMER-II calculations has a broad range extending from one- and two-dimensional, single subassembly calculations to two-dimensional, work-energy partition calculations for a complete pressure vessel.

B. Fluid-Dynamics and Structural Models

The fluid-dynamics and structural models in SIMMER-II are used to calculate the motion and thermodynamic behavior of the materials in LMFBR hypothetical core-disruptive accident (HCDA) sequences. The material distributions and temperatures predicted by these models affect the system reactivity as calculated with the neutronics models. The neutronics models in turn predict the energy production rate caused by fission and neutron capture heating in the reactor materials.

Fig. 1.
SIMMER-II cylindrical geometry.

A three-field approach is used to model the dynamic behavior of materials. All the materials in each "field" have the same velocity distribution; each velocity distribution is the solution to a set of conservation, or field, equations for mass, momentum, and energy. Each field contains several components, which are described by density and specific internal-energy distributions.

To account for different enrichment zones and different degrees of burnup, the fuel material in SIMMER-II is separated into a fertile and a fissile component. For example, in an LMFBR fueled with a mixed uranium- and plutonium-oxide, the fertile component can be considered to be depleted uranium dioxide and the fissile component to be primarily plutonium dioxide. These two components are modeled by two distinct density distributions, but because they are mixed intimately, only one energy distribution is necessary to determine their common local temperature. A control-material component, usually boron carbide

(B₄C), has been added to include the neutron absorption effect of this material and its thermal interactions with other materials in the reactor core. A fission-gas component also is included, not only to provide pressure when it is released from the unstructured fuel matrix to the vapor field but also to account for the presence of noncondensable gas in the vaporization-condensation models. Thus, SIMMER-II has six basic components: fertile fuel, fissile fuel, steel, sodium, control, and fission gas. However, as explained below, each field contains more than these six components.

Molten fuel that moves axially from the active core to the blankets can freeze in the blanket regions. In SIMMER-II, the densities and energies of the frozen-fuel components could have been combined with the densities and energies of blanket, fabricated-fuel components (almost all fertile) to obtain the new fuel-component average densities and energies. Because of the high mass of blanket material and its lower temperature, the average fuel temperature in this new mixture would have been low and thus would have produced an unrealistic state for the frozen fuel. To avoid this problem, SIMMER-II calculates separate density and energy distributions for both fabricated and frozen fuel in the structure field. Also, any fuel freezing as a result of liquid-to-pin heat transfer is assumed to stay in the liquid field as fuel particles; that is, stable crusts are assumed not to form on fuel pins. The control material is another structural material that could have been considered as a fabricated component and a frozen component. However, SIMMER-II attempts to consider the dynamics of the control material during disassembly and transition phases. Because this dynamic behavior is not well known, only a single structural control component is considered until analyses and equation-of-state information are available to better define the role of control material in HCDA sequences.

The concentration of fission gas in the fabricated fuel is modeled by the fission-gas densities in the structure field. The fission-gas temperature is assumed to be the same as the fabricated-fuel temperature. The fission gas in the structure field currently is assumed to be dissolved within the unstructured fuel matrix in two concentrations: intragranular and intergranular fission gas. No fission gas is assumed to remain within frozen fuel. The structure field in SIMMER-II also contains a steel-cladding component and a steel subassembly can-wall component. In summary, the structure field of SIMMER-II contains nine density components: fertile fuel in fabricated pellets, fissile fuel in fabricated pellets, frozen fertile fuel, frozen fissile fuel, steel cladding, steel subassembly can-wall, control material, intergranular fission gas, and

intragranular fission gas. The temperatures of these components are determined by the five specific internal energies for the following energy components: fabricated fuel, frozen fuel, cladding, subassembly can-wall, and control material.

With the exception of fission gas, the liquid field models the liquid state of the basic components of SIMMER-II. In addition, the liquid field of SIMMER-II contains solid fuel particles and solid steel particles. In summary, the liquid field contains eight density components: liquid fertile fuel, liquid fissile fuel, liquid steel, liquid sodium, liquid control material, fertile fuel in solid particles, fissile fuel in solid particles and steel in solid particles. The temperatures of the liquid-field components are determined by the six specific internal energies for the following energy components: liquid fuel, liquid steel, liquid sodium, liquid control, solid fuel particles, and solid steel particles.

The vapor field of SIMMER-II comprises the vapors of the six basic components: fertile fuel, fissile fuel, steel, sodium, control, and fission gas. The mixture of these components has a single local energy.

The field equations for the three fields are described in the following sections.

1. Subscript Notation. Because a one-to-one correspondence between the component densities and energies does not exist, the subscript notation differs in application between density and energy components. For density components, the lower-case subscripts *s*, *l*, and *g* refer to the structure, liquid, and vapor fields, respectively. The upper-case subscripts *S*, *L*, and *G* likewise are used for the energy components. These apply regardless of the physical state of a field component; hence, solid fuel particles in the liquid field are still subscripted with an *l* (or *L*). Furthermore, in SIMMER-II the fertile and fissile fuel components are assumed to have the same equation of state. Thus, in the vapor field, the two fuel-density components are combined into a single material component for use in the vaporization-condensation model. Again, to distinguish the material component from the density component, the upper-case subscript is used for the material component. Confusion with the energy-component subscript is avoided because numerical subscripts needed for the vapor-field material components are not required for the single, vapor-field, energy component. Overall field quantities, such as velocities, are subscripted with the upper-case subscripts. For consistency with the consecutive numbering scheme within the computer code itself, the density and energy components are numbered consecu-

tively in this manual. The density and energy component subscripts are listed in Table I.

This can be illustrated as follows. The fabricated fertile-fuel and fissile-fuel macroscopic densities (defined in Sec. III.B.2) are denoted by $\bar{\rho}_{s1}$ and $\bar{\rho}_{s2}$, respectively, and the fabricated-fuel specific internal energy is denoted by e_{s1} . To obtain the fabricated-fuel energy density needed for the energy equation, the total macroscopic density of fabricated fuel and the specific internal energy are multiplied. This density is the sum of $\bar{\rho}_{s1}$ and $\bar{\rho}_{s2}$, which is denoted by $\bar{\rho}_{S1}$. The equivalences between the component densities and densities that multiply the specific internal energies are indicated in Table II.

In SIMMER-II, an equation of state is necessary for each of the five materials. As indicated previously, a given material has several energy components. Thus, for accident-analysis calculations using the five materials, the material assignments shown in Table III are made. The internal ordering of the equations of state for the materials in SIMMER-II is

- (1) fuel,
- (2) steel,
- (3) sodium,
- (4) control, and
- (5) fission gas.

When an equation-of-state quantity is required for an energy (or material) component, an upper-case M is used as a subscript on the symbol for the quantity to indicate the transformation M(m), where m is the energy component number and M is the equation-of-state material number. The (m) generally is omitted for simplicity. Some confusion may be avoided by observing that components contributing to the vapor pressure possess identical M and m values. These include the five vapor material components and the first four liquid energy components. This correspondence does not hold for the solid particles and structure components.

Nomenclature is introduced throughout this manual as the need arises. For convenience, a complete list of the nomenclature is included in Appendix A.

2. Volume Fraction Treatment. In SIMMER-II, the macroscopic (smeared) density of a component is distinguished from the microscopic (normal or equation-of-state) density by placing a bar over the normal density symbol, as in $\bar{\rho}_{sm}$. The ratio of these two densities determines the volume fraction of the desired component. However, because of numerical expediency, the single-phase/two-phase transition is assumed to take place with a small residual vapor

TABLE I
DENSITY AND ENERGY COMPONENT SUBSCRIPTS

	<u>Density-Component Subscripts</u>		<u>Energy- and Material- Component Subscripts</u>	
	<u>Subscript</u>	<u>Component</u>	<u>Subscript</u>	<u>Component</u>
Structure Field	s1	Fabricated Fertile Fuel	S1	Fabricated Fuel
	s2	Fabricated Fissile Fuel		
	s3	Frozen Fertile Fuel	S2	Frozen Fuel
	s4	Frozen Fissile Fuel		
	s5	Cladding	S3	Cladding
	s6	Subassembly Can Wall	S4	Subassembly Can-Wall
	s7	Solid Control	S5	Solid Control
	s8	Intragranular Fission Gas		
	s9	Intergranular Fission Gas		
Liquid Field	L1	Liquid Fertile Fuel	L1	Liquid Fuel
	L2	Liquid Fissile Fuel		
	L3	Liquid Steel	L2	Liquid Steel
	L4	Liquid Sodium	L3	Liquid Sodium
	L5	Liquid Control	L4	Liquid Control
	L6	Solid Fertile Fuel Particles	L5	Solid Fuel Particles
	L7	Solid Fissile Fuel Particles		
	L8	Solid Steel Particles	L6	Solid Steel Particles
Vapor Field	g1	Fertile Fuel Vapor	G1	Fuel Vapor
	g2	Fissile Fuel Vapor		
	g3	Steel Vapor	G2	Steel Vapor
	g4	Sodium Vapor	G3	Sodium Vapor
	g5	Control Vapor	G4	Control Vapor
	g6	Fission Gas	G5	Fission Gas

TABLE II
DENSITY EQUIVALENCES FOR ENERGY COMPONENTS

<u>Component</u>	<u>Equivalence</u>
Structure Field	
Fabricated Fuel	$\bar{\rho}_{S1} = \bar{\rho}_{s1} + \bar{\rho}_{s2}$
Frozen Fuel	$\bar{\rho}_{S2} = \bar{\rho}_{s3} + \bar{\rho}_{s4}$
Cladding	$\bar{\rho}_{S3} = \bar{\rho}_{s5}$
Subassembly Can-Wall	$\bar{\rho}_{S4} = \bar{\rho}_{s6}$
Solid Control	$\bar{\rho}_{S5} = \bar{\rho}_{s7}$
Liquid Field	
Liquid Fuel	$\bar{\rho}_{L1} = \bar{\rho}_{\ell1} + \bar{\rho}_{\ell2}$
Liquid Steel	$\bar{\rho}_{L2} = \bar{\rho}_{\ell3}$
Liquid Sodium	$\bar{\rho}_{L3} = \bar{\rho}_{\ell4}$
Liquid Control	$\bar{\rho}_{L4} = \bar{\rho}_{\ell5}$
Solid Fuel Particles	$\bar{\rho}_{L5} = \bar{\rho}_{\ell6} + \bar{\rho}_{\ell7}$
Solid Steel Particles	$\bar{\rho}_{L6} = \bar{\rho}_{\ell8}$
Vapor Field	
Vapor Mixture	$\bar{\rho}_G = \bar{\rho}_g = \sum_{m=1}^{NCGR} \bar{\rho}_{gm}$
Fuel Vapor	$\bar{\rho}_{G1} = \bar{\rho}_{g1} + \bar{\rho}_{g2}$
Steel Vapor	$\bar{\rho}_{G2} = \bar{\rho}_{g3}$
Sodium Vapor	$\bar{\rho}_{G3} = \bar{\rho}_{g4}$
Control Vapor	$\bar{\rho}_{G4} = \bar{\rho}_{g5}$
Fission Gas	$\bar{\rho}_{G5} = \bar{\rho}_{g6}$

TABLE III
COMPONENT EQUATION-OF-STATE ASSIGNMENTS

<u>Structure Energy Component</u>	<u>Material Number</u>	<u>Material</u>
Fabricated Fuel	1	Fuel
Frozen Fuel	1	Fuel
Cladding	2	Steel
Can-Wall	2	Steel
Solid Control	4	Control
<u>Liquid Energy Component</u>		
Liquid Fuel	1	Fuel
Liquid Steel	2	Steel
Liquid Sodium	3	Sodium
Liquid Control	4	Control
Solid Fuel Particles	1	Fuel
Solid Steel Particles	2	Steel
<u>Vapor Density Component</u>		
Fertile Fuel Vapor	1	Fuel
Fissile Fuel Vapor	1	Fuel
Steel Vapor	2	Steel
Sodium Vapor	3	Sodium
Control Vapor	4	Control
Fission Gas	5	Fission Gas

volume. This vapor volume is designated as α_0 when no structure is present. (A more complete description is in Appendix B.) To allow the macroscopic liquid density to approach the microscopic density and to allow for the required void volume when structure is transformed into liquid, the liquid volume fractions are multiplied by $(1 - \alpha_0)$. Consequently, the volume fractions for the structure and liquid fields are given by

$$\alpha_{sm} = \frac{\bar{\rho}_{sm}}{\rho_{sm}}, \quad m = 1, 2, \dots, \text{NCSR}, \text{ and} \quad (\text{III-1})$$

$$\alpha_{lm} = \bar{\rho}_{lm} (1 - \alpha_0) / \rho_{lm} \quad m = 1, 2, \dots, \text{NCLR}, \quad (\text{III-2})$$

where

α_{sm} is the volume fraction of structure-field density component m ,

$\bar{\rho}_{sm}$ is the macroscopic density of structure-field density component m ,

ρ_{sm} is the microscopic density of structure-field density component m ,

α_{lm} is the volume fraction of liquid-field density component m ,

$\bar{\rho}_{lm}$ is the macroscopic density of liquid-field density component m ,

ρ_{lm} is the microscopic density of liquid-field density component m ,

α_0 is the fraction of the available flow volume determining the transition between two-phase and single-phase flow,

$\text{NCSR} = 9$ is the number of density components in the structure field, and

$\text{NCLR} = 8$ is the number of density components in the liquid field.

In the current version of SIMMER-II, only the first three liquid-field materials are allowed temperature-dependent microscopic densities. This translates to a temperature dependence for the first four liquid density components of Table I. The remaining microscopic densities are assumed to be time and

space independent. However, the transfer of mass from a solid-energy component to a liquid-energy component is based on a failure model in which the solid-component energy can exceed the solidus energy significantly before failure. That is, each solid-energy component can have sufficient energy such that a considerable portion of the component is molten. Most materials have approximately a 10% change in specific volume at the melting transition and SIMMER-II can account for the volume change. To do this, the microscopic densities used in Eqs. (III-1) and (III-2) are effective microscopic densities that are defined in the following manner. For each structure-field energy component, the melt fraction of the component is

$$f_{Sm} = \begin{cases} 1 & \text{if } e_{Sm} \geq e_{Liq,M} , \\ \frac{e_{Sm} - e_{Sol,M}}{h_{f,M}} & \text{if } e_{Liq,M} > e_{Sm} > e_{Sol,M} , \text{ and} \\ 0 & \text{if } e_{Sm} \leq e_{Sol,M} , \end{cases} \quad (III-3)$$

$m = 1, \dots, NCSE ,$

where

f_{Sm} is the fraction of structure-field energy component m that is molten,

e_{Sm} is the specific internal energy of structure-field energy component m ,

$e_{Sol,M}$ is the solidus energy of material M assigned to structure-field energy component m ,

$e_{Liq,M}$ is the liquidus energy of material M assigned to structure-field energy component m ,

$h_{f,M}$ is the heat of fusion for the same material M , and

$NCSE = 5$ is the number of energy components in the structure field.

Similarly, the fraction of each liquid-field solid energy component that is molten is given by

$$f_{Lm} = \begin{cases} 1 & \text{if } e_{Lm} \geq e_{Liq,M} \\ \frac{e_{Lm} - e_{Sol,M}}{h_{f,M}} & \text{if } e_{Liq,M} > e_{Lm} > e_{Sol,M} \text{ , and} \\ 0 & \text{if } e_{Lm} \leq e_{Sol,M} \end{cases} \quad (III-4)$$

$m = 5, 6 \text{ ,}$

where

f_{Lm} is the fraction of liquid-field energy component m that is molten and

e_{Lm} is the specific internal energy of liquid-field energy component m .

The microscopic densities used in Eqs. (III-1) and (III-2) then are

$$\frac{1}{\rho_{sm}} = \frac{f_{sm}}{\rho_{s\ell m}} + \frac{(1 - f_{sm})}{\rho_{ssm}} \text{ ,} \quad m = 1, \dots, NCSE \text{ , and} \quad (III-5)$$

$$\frac{1}{\rho_{\ell m}} = \frac{f_{\ell m}}{\rho_{\ell\ell m}} + \frac{(1 - f_{\ell m})}{\rho_{\ell sm}} \text{ ,} \quad m = 6, 7, 8 \text{ ,} \quad (III-6)$$

where

f_{sm} is the melt fraction of structure-field density component m ,

$\rho_{s\ell m}$ is the microscopic density of the pure liquid phase of structure-field density component m ,

ρ_{ssm} is the microscopic density of the pure solid phase of structure-field density component m ,

$f_{\ell m}$ is the melt fraction of liquid-field density component m ,

$\rho_{\ell\ell m}$ is the microscopic density of the pure liquid phase of liquid-field density component m, and

$\rho_{\ell sm}$ is the microscopic density of the pure solid phase of liquid-field density component m.

To allow added flexibility, all microscopic densities on the right-hand side of Eqs. (III-5) and (III-6) are input. It is at the user's discretion whether $\rho_{s\ell m}$ and $\rho_{\ell\ell m}$ for fuel and steel will agree with the temperature-dependent $\rho_{\ell m}$ at the melting temperature. Of course, in these equations the melt fraction for the density components is the same as for its corresponding energy component.

The sum of the volume fractions of structure components does not account for all the space unavailable for liquid or vapor flow. The fission-gas plenum portion of the fuel pin occupies volume, but this volume is not available for flow until the cladding fails. This nonflow volume fraction, denoted by α_{nf} , is input separately as a regionwise parameter and is included as a structure volume fraction until the cladding is predicted to fail. A nonflow region that can be moved as a function of time also is used to model movement of a plug in the reactor head, as is described in Secs. III.D.5.f and IV.D. Finally, the intersub-assembly gap volume can be modeled as a nonflow volume fraction denoted by $\alpha_{nf,cw}$ until radial motion is allowed. Thus, the total volume fraction of the structure field is

$$\alpha_S = \sum_{m=1}^{NCSR} \alpha_{sm} + \frac{\alpha_{nf} \min[\bar{p}_{s5}, (\bar{p}_{s5})_{fail}]}{(\bar{p}_{s5})_{fail}} + \alpha_{nf,cw} \quad , \quad (III-7)$$

if radial motion is not allowed at the right-hand boundary of a cell, and

$$\alpha_S = \sum_{m=1}^{NCSR} \alpha_{sm} + \frac{\alpha_{nf} \min[\bar{p}_{s5}, (\bar{p}_{s5})_{fail}]}{(\bar{p}_{s5})_{fail}} \quad , \quad (III-8)$$

if radial motion is allowed at the right-hand boundary of a cell. In Eqs. (III-7) and (III-8),

α_S is the total volume fraction of the structure field and

$(\bar{\rho}_{SS})_{fail}$ is the minimum cladding macroscopic density above which cladding is assumed to be intact.

The cladding nonflow volume fraction is removed gradually from the structure-field volume fraction after cladding failure to reduce the effect of sudden area changes on the fluid-dynamics calculation. The volumes represented by α_{nf} currently are assumed to be evacuated. Eventually such volumes (for example, the fission-gas plena) can be used to store nonmoving materials until the failure criterion is met. These materials then can be released or modeled to interact with the fluid fields through an appropriate algorithm. An alternative method of treating intersubassembly gaps is given in the structure configuration model description in Sec. III.D-1.

The total volume fraction of the liquid field is obtained by summing over its components,

$$\alpha_L = \sum_{m=1}^{NCLR} \alpha_{Lm} = \sum_{m=1}^{NCLR} \bar{\rho}_{Lm}(1 - \alpha_0)/\rho_{Lm} \quad . \quad (III-9)$$

The components of the vapor field are assumed to be compressible and thus capable of expanding to fill the remaining available volume. Hence, the volume fraction of the vapor field is that not taken by the structure and liquid fields:

$$\alpha_G = 1 - \alpha_S - \alpha_L \quad . \quad (III-10)$$

The vapor components also are assumed to be mixed completely such that each component expands to fill the available vapor volume. Thus, for the vapor components, the microscopic and macroscopic densities are related by

$$\bar{\rho}_{gm} = \rho_{gm} \alpha_G \quad , \quad m = 1, \dots, NCGR \quad , \quad (III-11)$$

where

$\bar{\rho}_{gm}$ is the macroscopic density of vapor-field density component m,

ρ_{gm} is the microscopic density of vapor-field density component m, and,

$N_{GFR} = 6$ is the number of density components in the vapor field.

3. The Structure Field. The structure field components model the materials that are stationary during at least one time step of an accident sequence. This includes materials that freeze and adhere during the accident as well as intact structural components before their possible failure and breakup. In SIMMER-II, the field equations for the structure field reduce to simple "point" equations because there is no motion of the structure field, and energy transfer between mesh cells is limited to special cases involving only the radial can-wall. Other codes¹⁻³ model such structural components in greater detail; however, as explained in Chap. II the modeling philosophy of SIMMER-II avoided similar treatments. The use of detailed exchange-function models to include LMFBR accident phenomena in the calculations partially compensates for the simplicity of the structure-field component equations. These exchange functions allow detail to be included in the calculation without complicating the field equations themselves.

The time variation of the structure mass is determined by the structure-field, density-component mass conservation equation:

$$\frac{\partial \bar{\rho}_{sm}}{\partial t} = -\Gamma_{sm} \quad , \quad m = 1, \dots, N_{CSR} \quad , \quad (III-12)$$

where Γ_{sm} is the total mass-transfer rate per unit volume from structure-field density component m. Throughout this report, "mass-transfer rate" is defined as mass-transfer rate per unit volume. A positive Γ_{sm} indicates that mass is being lost from component m. In general, this loss represents gains to components in the liquid field and represents breakup or melting of the structure-field component. For negative Γ_{sm} , the mass transfer results from the freezing and adhesion of liquid-field components to form structure-field component m.

As shown in Fig. 2, a number of mass-transfer paths exist in SIMMER-II. The Γ_{sm} of Eq. (III-12) represents the sum of all mass-transfer rates for structure-field component m. To distinguish further the individual mass-transfer rate for a given path, we define $\Gamma_{sm \leftarrow k}$, which is the mass-transfer rate per unit volume from structure-field component m to liquid-field component k. Although

Fig. 2.
SIMMER-II mass-transfer paths.

Γ_{sm} can be either positive or negative, $\Gamma_{sm\ell k}$ is always positive. If the mass transfer is from the liquid-field density component to the structure-field density component, then the mass-transfer rate is indicated by $\Gamma_{\ell ms k}$. That is, the mass source is indicated by the first subscript and component number, and the mass sink is indicated by the second subscript and component number. All individual mass-transfer rates for the structure-field density components are indicated in Table IV. The signs on the individual mass-transfer rates indicate how they contribute to the total mass-transfer rate.

Several assumptions were made in formulating the mass-transfer paths of Fig. 2 and Table IV. First, as the description of the structure-failure model indicates in Sec. III.D, fabricated fuel can break up and enter the liquid field while a portion of the fuel is still solid. The solid portion of the fabricated fuel that fails is transferred to the solid-fuel-particle component in the liquid field. Second, mass transfer from the fabricated fuel occurs in only one direction; that is, fabricated fuel cannot be remade from the freezing of liquid fuel. Such freezing mass transfer occurs only to the frozen fuel component. Similarly, the solid-fuel particles cannot return to the structure field although mechanical jamming of moving particles is modeled in SIMMER-II. Breakup of frozen fuel, cladding, and can-wall is treated similarly to that of the fabricated fuel. However, because no frozen-steel component is considered in SIMMER-II, the frozen steel that adheres is assumed to return to the cladding and can-wall components.

The mass-transfer rate from solid-control material in the structure field to liquid-control material in the liquid field is based only on the rate of melting. The fission-gas mass transfer remains within the structure field for mass transfer of intragranular fission gas to intergranular fission gas. Again, this is a one-way process; intergranular gas cannot return to the interior of the fuel grains.

The mass-transfer rate for each of the fertile- and fissile-fuel components is proportional to the total (fertile plus fissile) fuel mass-transfer rate for the given mass-transfer path. Partitioning the energy-component mass-transfer rate into the individual density-component mass-transfer rates is based on the mass fraction of fertile or fissile fuel in the fuel component contributing the mass.

The structure field remains stationary and thus has no momentum equation. However, the presence of the structure components does influence the fluid motion of the liquid and vapor fields. First, it restricts radial motion when

TABLE IV
STRUCTURE-FIELD MASS-TRANSFER RATES

<u>Total Mass-Transfer Rate</u>	<u>Individual Mass-Transfer Rate</u>	<u>Mass Source</u>	<u>Mass Sink</u>
Γ_{s1}	$+ \Gamma_{s1\ell1}$	Fabricated fertile fuel	Liquid fertile fuel
	$+ \Gamma_{s1\ell6}$	Fabricated fertile fuel	Fertile fuel in solid particles
Γ_{s2}	$+ \Gamma_{s2\ell2}$	Fabricated fissile fuel	Liquid fissile fuel
	$+ \Gamma_{s2\ell7}$	Fabricated fissile fuel	Fissile fuel in solid particles
Γ_{s3}	$+ \Gamma_{s3\ell1}$	Frozen fertile fuel	Liquid fertile fuel
	$+ \Gamma_{s3\ell6}$	Frozen fertile fuel	Fertile fuel in solid particles
	$- \Gamma_{\ell1s3}$	Liquid fertile fuel	Frozen fertile fuel
Γ_{s4}	$+ \Gamma_{s4\ell2}$	Frozen fissile fuel	Liquid fissile fuel
	$+ \Gamma_{s4\ell7}$	Frozen fissile fuel	Fissile fuel in solid particles
	$- \Gamma_{\ell2s4}$	Liquid fissile fuel	Frozen fissile fuel
Γ_{s5}	$+ \Gamma_{s5\ell3}$	Cladding	Liquid steel
	$+ \Gamma_{s5\ell8}$	Cladding	Solid steel particles
	$- \Gamma_{\ell3s5}$	Liquid steel	Cladding
Γ_{s6}	$+ \Gamma_{s6\ell3}$	Can-wall	Liquid steel
	$+ \Gamma_{s6\ell8}$	Can-wall	Solid steel particles
	$- \Gamma_{\ell3s6}$	Liquid steel	Can-wall
Γ_{s7}	$+ \Gamma_{s7\ell5}$	Solid control	Liquid control
	$- \Gamma_{\ell5s7}$	Liquid control	Solid control
Γ_{s8}	$+ \Gamma_{s8s9}$	Intragranular fission gas	Intergranular fission gas
Γ_{s9}	$+ \Gamma_{s9g6}$	Intergranular fission gas	Free fission gas
	$- \Gamma_{s8s9}$	Intragranular fission gas	Intergranular fission gas

subassembly can-wall is present and has not been predicted to fail. Second, the structure components influence the fluid motion through the momentum-exchange functions that model the drag forces exerted on the liquid and vapor fields. These momentum-exchange functions are described in Sec. III.D.

The energy equations for the structure-field energy components have the form

$$\frac{\partial \bar{p}_{Sm} e_{Sm}}{\partial t} = Q_{Sm} + N Q_{NSm} , \quad m = 1, \dots, NCSE , \quad (III-13)$$

where

$\bar{p}_{Sm}$ is the macroscopic density for structure-field energy component m ,

e_{Sm} is the specific internal energy for structure-field energy component m ,

Q_{Sm} is the total energy-transfer rate per unit volume to structure-field energy component m ,

N is the instantaneous relative amplitude of the internal energy-generation rate,

Q_{NSm} is the normalized (relative to the amplitude N) internal energy-generation rate per unit volume for structure-field energy component m , and

$NCSE = 5$ is the number of structure-field energy components.

For a given structure-field energy component, the normalized internal energy-generation rate per unit volume is the sum of the normalized specific rates for the density components,

$$Q_{NSm} = \sum_{k \in m} \bar{p}_{sk} Q_{Nsk} , \quad (III-14)$$

where the summation is over all density components constituting energy component m and Q_{Nsk} is the normalized, internal, energy-generation rate per unit mass of density component k .

The total energy-transfer rate Q_{Sm} accounts for all energy transfer except for internal energy generation:

$$Q_{Sm} = q_{HSm} + q_{rSm} + q_{KSm} \quad , \quad (III-15)$$

where

q_{HSm} is the total heat-transfer rate to structure-field energy component m ,

q_{rSm} is the total energy transfer from mass transfer to structure-field energy component m , and

q_{KSm} is the total energy transfer to structure-field energy component m from drag heating.

The number of possible heat-transfer paths contributing to q_{HSm} is greater than the number of mass-transfer paths indicated in Fig. 2. The paths generally include heat transfer between two structure-field energy components, between a structure-field energy component and a liquid-field energy component, and between a structure-field energy component and the vapor field. Because of the large number of possible paths, they are not tabulated. However, we do note that opposite subscript conventions from the mass-transfer paths apply for energy transfer; for example, q_{HSmLk} denotes the heat-transfer rate to structure-field energy component m from liquid-field energy component k . However, heat does not always flow directly to another component. Both the nonequilibrium melting-freezing model and the vaporization-condensation model only involve heat flow from an interface. For melting/freezing, this contribution to q_{HSm} will be called q_{ISmL} , the heat flow from the liquid-field/structure interface to structure component m . For vaporization-condensation, q_{ISmGk} will be identified as the heat flow from the structure-vapor interface to structure component m because of vapor component k . Again, because energy is thought of as flowing to a given component, the first subscripts identify the destination of the energy flow. The second subscript(s) identify the component(s) influencing the interfacial temperature.

4. The Liquid Field. The liquid field contains those components that are mobile and are in the liquid or solid state. These components are assumed to have the same instantaneous velocity distribution; that is, no relative motion exists between the liquid-field components.

Mass conservation for the liquid-field density components is determined by

$$\frac{\partial \bar{\rho}_{\ell m}}{\partial t} + \nabla \cdot (\bar{\rho}_{\ell m} \vec{v}_L) = -\Gamma_{\ell m} \quad , \quad m = 1, \dots, NCLR \quad , \quad (\text{III-16})$$

where

$\bar{\rho}_{\ell m}$ is the macroscopic density of liquid-field density component m ;

$\vec{v}_L$ is the liquid-field velocity vector with components U_L and V_L in the directions r (or x) and z , respectively;

$$\begin{aligned} \nabla \cdot (\bar{\rho}_{\ell m} \vec{v}_L) &= \frac{1}{r} \frac{\partial}{\partial r} (r \bar{\rho}_{\ell m} U_L) + \frac{\partial}{\partial z} (\bar{\rho}_{\ell m} V_L) \text{ for } (r, z) \text{ geometry, or} \\ &= \frac{\partial}{\partial x} (\bar{\rho}_{\ell m} U_L) + \frac{\partial}{\partial z} (\bar{\rho}_{\ell m} V_L) \text{ for } (x, z) \text{ geometry; and} \end{aligned}$$

$\Gamma_{\ell m}$ is the total mass-transfer rate per unit volume from liquid-field density component m .

Again, a positive $\Gamma_{\ell m}$ indicates that mass is being lost from density component m . As shown in Fig. 2, the possible mass-transfer paths to the liquid-field components include mass exchange with the vapor field as well as with the structure field. Table V lists the individual contributions to the total mass-transfer rates in Eq. (III-16).

For the liquid-field momentum equation, we define the total liquid-field density as

$$\bar{\rho}_L = \sum_{m=1}^{NCLR} \bar{\rho}_{\ell m} \quad . \quad (\text{III-17})$$

TABLE V
LIQUID-FIELD MASS-TRANSFER RATES

<u>Total</u> <u>Mass-Transfer</u>	<u>Individual</u> <u>Mass-Transfer</u>	<u>Source</u>	<u>Sink</u>
$\Gamma_{\ell 1}$	+ $\Gamma_{\ell 1 s 3}$	Liquid fertile fuel	Frozen fertile fuel
	+ $\Gamma_{\ell 1 \ell 6}$	Liquid fertile fuel	Fertile fuel in solid particles
	+ $\Gamma_{\ell 1 g 1}$	Liquid fertile fuel	Fertile fuel vapor
	- $\Gamma_{s 1 \ell 1}$	Fabricated fertile fuel	Liquid fertile fuel
	- $\Gamma_{s 3 \ell 1}$	Frozen fertile fuel	Liquid fertile fuel
	- $\Gamma_{\ell 6 \ell 1}$	Fertile fuel in solid particles	Liquid fertile fuel
	- $\Gamma_{g 1 \ell 1}$	Fertile fuel vapor	Liquid fertile fuel
$\Gamma_{\ell 2}$	+ $\Gamma_{\ell 2 s 4}$	Liquid fissile fuel	Frozen fissile fuel
	+ $\Gamma_{\ell 2 \ell 7}$	Liquid fissile fuel	Fissile fuel in solid particles
	+ $\Gamma_{\ell 2 g 2}$	Liquid fissile fuel	Fissile fuel vapor
	- $\Gamma_{s 2 \ell 2}$	Fabricated fissile fuel	Liquid fissile fuel
	- $\Gamma_{s 4 \ell 2}$	Frozen fissile fuel	Liquid fissile fuel
	- $\Gamma_{\ell 7 \ell 2}$	Fissile fuel in solid particles	Liquid fissile fuel
	- $\Gamma_{g 2 \ell 2}$	Fissile fuel vapor	Liquid fissile fuel
$\Gamma_{\ell 3}$	+ $\Gamma_{\ell 3 s 5}$	Liquid steel	Cladding
	+ $\Gamma_{\ell 3 s 6}$	Liquid steel	Can-wall
	+ $\Gamma_{\ell 3 \ell 8}$	Liquid steel	Solid steel particles
	+ $\Gamma_{\ell 3 g 3}$	Liquid steel	Steel vapor
	- $\Gamma_{s 5 \ell 3}$	Cladding	Liquid steel
	- $\Gamma_{s 6 \ell 3}$	Can-wall	Liquid steel
	- $\Gamma_{\ell 8 \ell 3}$	Solid steel particles	Liquid steel
	- $\Gamma_{g 3 \ell 3}$	Steel vapor	Liquid steel
$\Gamma_{\ell 4}$	+ $\Gamma_{\ell 4 g 4}$	Liquid sodium	Sodium vapor
	- $\Gamma_{g 4 \ell 4}$	Sodium vapor	Liquid sodium

TABLE V (CONT)

<u>Total</u> <u>Mass Transfer</u>	<u>Individual</u> <u>Mass Transfer</u>	<u>Mass</u> <u>Source</u>	<u>Mass</u> <u>Sink</u>
Γ_{15}	$+ \Gamma_{15s7}$	Liquid control	Solid control
	$+ \Gamma_{15g5}$	Liquid control	Control vapor
	$- \Gamma_{s715}$	Solid control	Liquid control
	$- \Gamma_{g515}$	Control vapor	Liquid control
Γ_{16}	$+ \Gamma_{1611}$	Fertile fuel in solid particles	Liquid fertile fuel
	$- \Gamma_{s116}$	Fabricated fertile fuel	Fertile fuel in solid particles
	$- \Gamma_{s316}$	Frozen fertile fuel	Fertile fuel in solid particles
	$- \Gamma_{1116}$	Liquid fertile fuel	Fertile fuel in solid particles
Γ_{17}	$+ \Gamma_{1712}$	Fissile fuel in solid particles	Liquid fissile fuel
	$- \Gamma_{s217}$	Fabricated fissile fuel	Fissile fuel in solid particles
	$- \Gamma_{s417}$	Frozen fissile fuel	Fissile fuel in solid particles
	$- \Gamma_{1217}$	Liquid fissile fuel	Fissile fuel in solid particles
Γ_{18}	$+ \Gamma_{1813}$	Solid steel particles	Liquid steel
	$- \Gamma_{s518}$	Cladding	Solid steel particles
	$- \Gamma_{s618}$	Can-wall	Solid steel particles
	$- \Gamma_{1318}$	Liquid steel	Solid steel particles

Also needed for the liquid-field momentum equation is the separation of the mass transferred from the liquid field and the mass transferred to the liquid field. Therefore, the following definitions are made.

$$\Gamma_{SL} = \sum_{m=1}^{NCSR} \sum_{k=1}^{NCLR} \Gamma_{sm-k} \quad , \quad (III-18)$$

$$\Gamma_{LS} = \sum_{k=1}^{NCLR} \sum_{m=1}^{NCSR} \Gamma_{k-sm} \quad , \quad (III-19)$$

$$\Gamma_{LG} = \sum_{m=1}^{NCLR} \Gamma_{\ell m g m} , \quad \text{and} \quad (III-20)$$

$$\Gamma_{GL} = \sum_{m=1}^{NCGR} \Gamma_{g m \ell m} , \quad (III-21)$$

where

Γ_{SL} is the total mass-transfer rate from the structure to the liquid field,

Γ_{LS} is the total mass-transfer rate from the liquid to the structure field,

Γ_{LG} is the total mass-transfer rate from the liquid to the vapor field and

Γ_{GL} is the total mass-transfer rate from the vapor to the liquid field.

Using the above definitions, the liquid-field momentum equation is

$$\begin{aligned} \frac{\partial \bar{p}_L}{\partial t} \vec{v}_L + \nabla \cdot (\bar{p}_L \vec{v}_L \vec{v}_L) = & -\alpha_L \nabla p + \bar{p}_L \vec{g} + \Gamma_{GL} \vec{v}_G - (\Gamma_{LS} + \Gamma_{LG}) \vec{v}_L \\ & + K_{GL} (\vec{v}_G - \vec{v}_L) - K_{LS} \vec{v}_L + \nabla \cdot (\alpha_L \vec{\pi}_L \cdot \vec{v}_L) , \end{aligned} \quad (III-22)$$

where

p is the local pressure,

$\vec{g}$ is the acceleration of gravity;

$\vec{v}_G$ is the vapor-field velocity vector with components U_G and V_G in the r (or x) and z directions, respectively;

K_{GL} is the momentum-exchange function that models the drag forces between the liquid and vapor fields; and

K_{LS} is the momentum-exchange function that models the drag force between the liquid and structure fields.

The components of the viscous drag term, $\nabla \cdot (\alpha_L \vec{\pi}_L \cdot \vec{v}_L)$, for the first four liquid-field energy components are

$$\nu_L \left[\frac{4}{3} \frac{\partial}{\partial r} \left(\bar{\rho}_L \frac{\partial}{\partial r} r u_L \right) + \frac{\partial}{\partial z} \left(\bar{\rho}_L \frac{\partial u_L}{\partial z} \right) + \frac{1}{3} \frac{\partial}{\partial z} \left(\bar{\rho}_L \frac{\partial v_L}{\partial r} \right) \right] , \text{ and}$$

$$\nu_L \left[\frac{4}{3} \frac{\partial}{\partial z} \left(\bar{\rho}_L \frac{\partial v_L}{\partial z} \right) + \frac{1}{r} \frac{\partial}{\partial r} \left(\bar{\rho}_L r \frac{\partial v_L}{\partial r} \right) + \frac{1}{3} \frac{1}{r} \frac{\partial}{\partial r} \left(\bar{\rho}_L r \frac{\partial u_L}{\partial z} \right) \right] ,$$

for the radial and axial components, respectively, in (r,z) geometry, and

$$\nu_L \left[\frac{4}{3} \frac{\partial}{\partial x} \left(\bar{\rho}_L \frac{\partial u_L}{\partial x} \right) + \frac{\partial}{\partial z} \left(\bar{\rho}_L \frac{\partial u_L}{\partial z} \right) + \frac{1}{3} \frac{\partial}{\partial z} \left(\bar{\rho}_L \frac{\partial v_L}{\partial x} \right) \right] , \text{ and}$$

$$\nu_L \left[\frac{4}{3} \frac{\partial}{\partial z} \left(\bar{\rho}_L \frac{\partial v_L}{\partial z} \right) + \frac{\partial}{\partial x} \left(\bar{\rho}_L \frac{\partial v_L}{\partial x} \right) + \frac{1}{3} \frac{\partial}{\partial x} \left(\bar{\rho}_L \frac{\partial u_L}{\partial z} \right) \right] ,$$

for the x and z components, respectively, in (x,z) geometry where ν_L is an input constant coefficient of kinematic viscosity for the liquid field. This viscous-drag term, available as an option in SIMMER-II, models the viscous shear between the liquid-field flows in adjacent mesh cells.

The pressure in Eq. (III-22) is evaluated locally from the equation of state. For single-phase flow, the liquid equation of state determines the pressure, whereas the vapor equation of state determines the pressure for two-phase flow. The liquid-field volume fraction is placed outside the pressure gradient and the viscous drag term of Eq. (III-22) so that liquid-field acceleration does not occur in the absence of a pressure gradient and other forces. The mass-transfer terms, which are present in Eq. (III-22), account for the momentum transfer that occurs when mass is transferred between fields of different velocities. Such a term also exists for mass transfer from the structure field to the liquid field, but because the structure has a zero velocity, the term is not present in Eq. (III-22). The effect of this latter structure-field momentum transfer is present in adding mass to the liquid field. The momentum-exchange functions K_{GL} and K_{LS} are described in detail in Sec. III.D. The liquid-vapor momentum exchange function K_{GL} could have a $(1 - \alpha_S)$ multiplier; however, this has been incorporated into the K_{GL} definition. The viscous drag formula is

highly approximate; a variation in viscosity as a function of composition and temperature has not been attempted because of the rather inviscid character of liquid reactor materials.

The energy equations for the liquid-field energy components have the form

$$\frac{\partial \bar{\rho}_{Lm} e_{Lm}}{\partial t} + \nabla \cdot (\bar{\rho}_{Lm} e_{Lm} \vec{v}_L) = Q_{Lm} + N Q_{NLm} ,$$

$$m = 1, \dots, NCLE , \quad (III-23)$$

where

$\bar{\rho}_{Lm}$ is the macroscopic density for liquid-field energy component m ,

e_{Lm} is the specific internal energy of liquid-field energy component m ,

Q_{Lm} is the total energy-transfer rate per unit volume to liquid-field energy component m ,

Q_{NLm} is the normalized internal energy generation rate for liquid-field energy component m , and

$NCLE = 6$ is the number of liquid-field energy components.

Again, the normalized, internal, energy-generation rate is the sum of the rates for the density components

$$Q_{NLm} = \sum_{k \in m} \bar{\rho}_{Lk} Q_{NLk} , \quad (III-24)$$

where the summation is over all density components that constitute energy component m , and Q_{NLk} is the normalized internal energy generation rate per unit mass of density component k .

The total energy-transfer rate consists of a heat-transfer term, a mass-transfer term, a drag-heating term, and a viscous-heating term,

$$Q_{Lm} = q_{Hm} + q_{\Gamma m} + q_{Km} + q_{Vm} , \quad (III-25)$$

where

- q_{HLm} is the total heat-transfer rate to liquid-field energy component m ,
- q_{rLm} is the total energy transfer as a result of mass transfer to liquid-field energy component m ,
- q_{KLm} is the total energy transfer to liquid-field energy component m because of drag heating, and
- q_{vLm} is the energy generation in liquid-field energy component m because of viscous heating resulting when the option to model shear flow between adjacent mesh cells is used.

Again, many heat-transfer paths contribute to q_{HLm} , and they are not tabulated. Possible paths include heat transfer between a liquid-field energy component and a structure-field energy component, between two liquid-field energy components, and between a liquid-field energy component and the vapor field. The previously discussed subscript notation is used to identify the path for a particular heat-transfer process. The two special cases for interfacial contributions involve heat transfer to a liquid component from a liquid/vapor interface and a structure/liquid interface.

5. The Vapor Field. The components of the vapor field model materials that have been vaporized or exist in a free gaseous state (for example, fission gas). These components can have a different velocity distribution from the liquid-field velocity distribution. However, all vapor-field components have the same velocity at any point in space.

Mass conservation for the vapor-field density components is ensured by the following equation:

$$\frac{\partial \bar{\rho}_{gm}}{\partial t} + \nabla \cdot (\bar{\rho}_{gm} \vec{V}_G) = -\Gamma_{gm} \quad , \quad m = 1, \dots, N_{CGR} \quad , \quad (III-26)$$

where

- $\bar{\rho}_{gm}$ is the macroscopic density of vapor-field density component m ,
and
- Γ_{gm} is the total mass-transfer rate per unit volume from vapor-field density component m .

As shown in Fig. 2, the mass-transfer rates to or from the vapor field result from vaporization or condensation. The single exception is the release of

intergranular fission gas from the solid fuel. Thus, for the first five vapor-field density components,

$$\Gamma_{gm} = \Gamma_{gm\ell m} - \Gamma_{\ell m gm} \quad , \quad m = 1, \dots, 5, \quad (\text{III-27})$$

where the mass-transfer rates $\Gamma_{gm\ell m}$ and $\Gamma_{\ell m gm}$ are indicated in Table V. The sixth mass-transfer rate Γ_{g6} is equal to $-\Gamma_{s9g6}$ shown at the bottom of Table IV. Summation of the three mass-continuity equations, Eqs. (III-12), (III-16), and (III-26) for all density components gives zero for the right side of the final equation, thus conserving the system total mass.

Because the vapor-field density components all move with the same velocity, the total vapor-field density is necessary for the vapor-field momentum equation. This total density is given by

$$\bar{\rho}_G = \sum_{m=1}^{\text{NCGR}} \bar{\rho}_{gm} \quad . \quad (\text{III-28})$$

Then the vapor-field momentum equation is

$$\begin{aligned} \frac{\partial \bar{\rho}_G \vec{v}_G}{\partial t} + \nabla \cdot (\bar{\rho}_G \vec{v}_G \vec{v}_G) = & -\alpha_G \nabla p + \bar{\rho}_G \vec{g} + \Gamma_{LG} \vec{v}_L \\ & - \Gamma_{GL} \vec{v}_G + K_{GL} (\vec{v}_L - \vec{v}_G) - K_{GS} \vec{v}_G + \nabla \cdot (\alpha_G \vec{\pi}_G \cdot \vec{v}_G) \quad , \quad (\text{III-29}) \end{aligned}$$

where

K_{GS} is the momentum exchange function that models the drag force between the vapor and structure fields and

$\nabla \cdot (\alpha_G \vec{\pi}_G \cdot \vec{v}_G)$ models viscous drag in the vapor field and has a form similar to the liquid-field viscous drag of Eq. (III-22).

Adding the two momentum equations [Eqs. (III-22) and (III-29)] yields a nonzero right side. The pressure gradients and gravitational terms remain to accelerate the liquid and vapor fields. In addition, the structure drag terms remain because the structure field is assumed to be infinitely rigid. The presence of

these source terms means that momentum is not conserved in SIMMER-II; for example, the structure field is an infinite momentum sink.

The components of the vapor field are assumed to be mixed completely at each point in space and to have a single energy, temperature, and pressure. The time variation of the vapor mixture energy is modeled by

$$\frac{\partial \bar{\rho}_G e_G}{\partial t} + \nabla \cdot (\bar{\rho}_G e_G \vec{v}_G) = -p \nabla \cdot (\alpha_G \vec{v}_G + \alpha_L \vec{v}_L) + Q_G + N \sum_{m=1}^{NCGR} \bar{\rho}_{gm} Q_{Ngm} \quad (III-30)$$

where

e_G is the specific internal energy of the vapor mixture,

Q_G is the total energy-transfer rate per unit volume to the vapor mixture, and

Q_{Ngm} is the normalized, internal, energy-generation rate per unit mass of density component m.

The first term on the right side of Eq. (III-30) is a pressure-volume work term derived by assuming the liquids to be incompressible. The last two terms are similar to terms in the liquid- and structure-field equations. The energy-transfer term Q_G again is separated into a heat-transfer term, q_{HG} ; a mass-transfer energy term, q_{TG} ; a drag-heating term, q_{KG} ; and a viscous heating term, q_{VG} . Although these terms do not necessarily possess component subscripts, the interfacial contributions to q_{HG} will depend on component energy transfers; for example, q_{IGmLm} is the energy transferred to the vapor field from the liquid-vapor interface of liquid component m.

Summation of all the energy equations [Eqs. (III-13), (III-23), and (III-30)] yields a nonzero right side. The internal energy-generation terms, pressure-volume work term, and drag-heating terms remain. However, the heat-transfer and mass-transfer energy terms do cancel to conserve energy during these processes.

C. Equation-of-State Models

The equations of motion presented in the previous section are not complete without the equations of state relating the microscopic densities and internal energies of the components in the three fields to the pressures and temperatures. Furthermore, equation-of-state information is required for the phase-transition models to predict the mass-transfer rates. Two types of equations of state are available in SIMMER-II. First, the analytic equation of state of SIMMER-I has been extended to include control material and fission gas as well as fuel, steel, and sodium. Second, tabular equations of state are included as an option to provide realistic material properties that can be revised as new supporting experimental data become available.

1. Analytic Equation of State. The analytic equations of state used in SIMMER-II are simplified models that are not sufficiently general to apply to all thermodynamic states. Several of the simplifications arise from assumptions made in the development of the fluid-dynamics methods. Other simplifications are caused by the analytic nature of the equations of state; this nature assists in understanding the thermodynamic phenomena occurring during an accident calculation. More complex equations of state do not have this advantage.

a. Material Temperatures. The thermodynamic relation between the temperature and the energy of a material is obtained by integrating the specific heat of the material over temperature from 0 K to the material temperature. For the solid state, the specific heats are assumed constant. Thus,

$$e_{Sm} = c_{vSM} T_{Sm} , \quad m = 1, \dots, NCSE , \quad (III-31)$$

where

c_{vSM} is the constant-volume specific heat for the solid phase of material M,

$M = M(m)$ is the equation-of-state material number for structure-field energy component m, and

T_{Sm} is the temperature of structure-field energy component m.

At the melting temperature, Eq. (III-31) yields the definition of the solidus energy:

$$e_{Sol,M} = c_{vSM} T_{Melt,M} , \quad M = 1, \dots, NMAT , \quad (III-32)$$

where

$e_{\text{Sol},M}$ is the solidus energy for material M,

$T_{\text{Melt},M}$ is the melting temperature of material M, and

$N_{\text{MAT}} = 5$ is the number of materials.

The material temperature is assumed to remain constant during the melting process as the energy equivalent to the material heat of fusion is absorbed. Completion of the melting process defines the liquidus energy,

$$e_{\text{Liq},M} = e_{\text{vSM}} T_{\text{Melt},M} + h_{f,M} , \quad M = 1, \dots, N_{\text{MAT}} , \quad (\text{III-33})$$

where

$e_{\text{Liq},M}$ is the liquidus energy of material M and

$h_{f,M}$ is the heat of fusion of material M.

The transition from the liquid state to the vapor state is similar to the solid-liquid transition. However, the transition temperature is the saturation temperature for the material, and the saturation temperature is dependent on the local pressure. In SIMMER-II, this vapor pressure-temperature relation is

$$p_{vM} = p_M^* e^{\left(\frac{-\Gamma_M^*}{T_{\text{Sat},M}}\right)} , \quad M = 1, \dots, N_{\text{MAT}} , \quad (\text{III-34})$$

where

p_{vM} is the vapor pressure for material M,

p_M^* and T_M^* are input parameters for material M, and

$T_{\text{Sat},M}$ is the saturation temperature for material M.

Equation (III-34) is inverted easily to yield the saturation temperature:

$$T_{\text{Sat},M} = \frac{T_M^*}{\ln \left(\frac{p_M^*}{p_{vM}} \right)} . \quad (\text{III-35})$$

The energy for each liquid material at the saturation temperature is obtained using a constant liquid heat capacity until the saturation temperature reaches two-thirds of the critical temperature. For higher temperatures, the average enthalpy between the liquid and vapor phases of a material (neglecting pressure-volume effects) is approximated by extrapolating linearly to the assumed critical point energy. Thus, the condensate energy is obtained by subtracting one-half the heat of vaporization from the average internal energy extrapolated to the critical energy. The condensate energy is defined by

$$e_{\text{Con},M} = e_{\text{Liq},M} + c_{\text{vLM}} (T_{\text{Sat},M} - T_{\text{Melt},M}) ,$$

$$T_{\text{Sat},M} \leq \frac{2}{3} T_{\text{Crt},M} ,$$

$$e_{\text{Con},M} = e_{\text{Liq},M} + c'_{\text{vLM}} (T_{\text{Sat},M} - \frac{2}{3} T_{\text{Crt},M}) + a_{\text{L},M} - \frac{1}{2} h_{\text{lg},M} \quad (\text{III-36})$$

$$T_{\text{Sat},M} > \frac{2}{3} T_{\text{Crt},M}$$

$$M = 1, \dots, \text{NMAT},$$

where

$e_{\text{Liq},M}$ is the liquidus energy of material M,

$h_{\text{lg},M}$ is the enthalpy of vaporization of material M,

c_{vLM} is the constant-volume specific heat for the liquid phase for material M,

$T_{\text{Sat},M}$ is the saturation temperature for material M,

$T_{\text{Crt},M}$ is the critical temperature for material M,

$T_{\text{Melt},M}$ is the melting temperature for material M,

$a_{\text{L},M}$ is defined to obtain continuity in $e_{\text{Con},M}$ at $T_{\text{Sat},M} = \frac{2}{3} T_{\text{Crt},M}$ and

c'_{vLM} is defined such that the input critical point energy, $e_{\text{Crt},M}$, is obtained.

The saturated vapor energy then is defined for each material by adding the heat of vaporization to the condensate energy and subtracting the work involved in the expansion process, or

$$e_{vap,M} = e_{Con,M} + h_{lg,M} - p\Delta v_M \quad , \quad M = 1, \dots, NMAT \quad . \quad (III-37)$$

Unlike the solidus and liquidus energies, the condensate energy and saturated-vapor energy vary with pressure because $T_{Sat,M}$ depends on the pressure. In addition, the heat of vaporization is a function of the saturation temperature:

$$h_{lg,M} = h_{lg,M}^* \left(1 - \frac{T_{Sat,M}}{T_{Crt,M}}\right)^{\xi_M} \quad , \quad M = 1, \dots, NMAT \quad , \quad (III-38)$$

where $h_{lg,M}^*$ and ξ_M are input parameters for material M. Thus, $a_{L,M}$ and c'_{vLM} are easily evaluated as

$$a_{L,M} = \frac{1}{2} h_{lg,M}^* \left(\frac{1}{3}\right)^{\xi_M} + c_{vLM} \left(\frac{2}{3} T_{Crt,M} - T_{Melt,M}\right) \quad , \quad \text{and} \quad (III-39)$$

$$c'_{vLM} = 3 \frac{(e_{Crt,M} - e_{Liq,M} - a_{L,M})}{T_{Crt,M}} \quad , \quad M = 1, \dots, NMAT \quad . \quad (III-40)$$

The pressure-volume work term in Eq. (III-37) is obtained by differentiating the vapor pressure relation [Eq. (III-34)] and then using the Clapeyron equation:

$$\frac{dp_{vM}}{dT_{Sat,M}} = \frac{h_{lg,M}}{T_{Sat,M} \Delta v_M} \quad . \quad (III-41)$$

Consequently, we obtain

$$p\Delta v_M = \frac{T_{Sat,M}}{T_M^*} h_{lg,M} \quad . \quad (III-42)$$

The equations for the liquid component internal energies are evaluated in a fashion similar to that involved in obtaining the condensate energies. They are

$$e_{Lm} = e_{Liq,M} + c_{vLM} (T_{Lm} - T_{Melt,M}) , \quad T_{Lm} \leq \frac{2}{3} T_{Crt,M} , \quad (III-43)$$

$$e_{Lm} = e_{Liq,M} + c'_{vLM} (T_{Lm} - \frac{2}{3} T_{Crt,M}) + a_{L,M} - \frac{1}{2} h_{lg,M}^+ , \quad T_{Lm} > \frac{2}{3} T_{Crt,M} ,$$

$$m = 1, \dots, 4 ,$$

where

$$h_{lg,M}^+ = h_{lg,M}^* (1 - \frac{T_{Lm}}{T_{Crt,M}})^{\xi_M} , \quad \text{for} \quad T_{Lm} < T_{Crt,M} ,$$

$$h_{lg,M}^+ = 0 , \quad \text{for} \quad T_{Lm} \geq T_{Crt,M} , \text{ and}$$

T_{Lm} is the temperature of liquid-field energy component m .

Only four liquid-field energy components are indicated for Eq. (III-43) because only the first four are in the liquid state. The solid-fuel and solid-steel energies are determined from Eq. (III-31). Addition of energy to the vapor increases its energy above saturation according to

$$e_{G,M} = e_{Vap,M} + c_{vGM} (T_G - T_{Sat,M}) \quad M = 1, \dots, NMAT , \quad (III-44)$$

where

c_{vGM} is the constant-volume specific heat of vapor phase of material M
and

T_G is the vapor temperature for the vapor mixture.

For fission gas ($M = 5$), the equation for the vapor energy is revised to neglect pressure dependence on the internal energy. The assumed relationship is

$$e_{G,5} = e_{Crt,5} + c_{vG5} (T_G - T_{Crt,5}) . \quad (III-45)$$

This is accomplished by always evaluating $e_{G,5}$ from

$$e_{G,5} = e_{Liq,5} + c'_{vL5} (T_{Sat,5} - \frac{2}{3} T_{Crt,5}) + a_{L,5} + \frac{1}{2} h_{lg,5} - p\Delta v_5 + c_{vG5} (T_G - T_{Sat,5}) \quad (III-46)$$

and presetting the parameters in this equation with the constants

$$c'_{vL5} = c_{vG5} ,$$

$$a_{L,5} = e_{Crt,5} - \frac{1}{3} c_{vG5} T_{Crt,5} - e_{Liq,5} ,$$

$$h_{lg,5} = 0 , \text{ and}$$

$$T_{Sat,5} = 10^4 .$$

Equation (III-44) is written for each material. In general, SIMMER-II computes a mixture of vapors having a single energy and temperature. To obtain a relationship between the vapor temperature and vapor energy, we define

$$e_{G,M}^* = e_{G,M} - c_{vGM} T_G , \quad M = 1, \dots, NMAT . \quad (III-47)$$

Then we assume that the total mixture energy is the sum of the energies of the mixture components, or

$$e_G = \sum_{m=1}^{NMAT} x_m e_{G,M} , \quad (III-48)$$

where $x_m = \frac{p_{Gm}}{p_G}$ is the mass fraction of vapor material m in the vapor mixture.

Because the same equation of state is used for fertile-fuel vapor and fissile-fuel vapor, their mass fractions have been combined to form x_1 . Substitution of Eq. (III-47) into Eq. (III-48) results in

$$e_G = T_G \left(\sum_{m=1}^{NMAT} x_m c_{vGM} \right) + \sum_{m=1}^{NMAT} x_m e_{G,M}^* . \quad (III-49)$$

Equation (III-31) is inverted easily for the structure and solid particle temperatures:

$$T_{Sm} = \frac{e_{Sm}}{c_{vSM}} , \quad m = 1, \dots, \text{NCSE and} \quad (\text{III-50})$$

$$T_{Lm} = \frac{e_{Lm}}{c_{vSM}} , \quad m = 5, 6 . \quad (\text{III-51})$$

Below $2/3 T_{\text{Crt},M}$, Eq. (III-43) can be inverted for T_{Lm} as

$$T_{Lm} = T_{\text{Melt},M} + \frac{(e_{Lm} - e_{\text{Liq},M})}{c_{vLM}} , \quad m = 1, 2, 3, 4 . \quad (\text{III-52})$$

Above $2/3 T_{\text{Crt},M}$, Eq. (III-43) is solved for T_{Lm} by a Newton-Raphson procedure as is explained in Sec. IV.C.2. Equation (III-49) contains the vapor temperature both explicitly and implicitly (through the pressure). Again, Sec. IV details the algorithm for solving for the vapor temperature.

b. Two-Phase Flow Pressures. For two-phase flow, the local pressure is obtained from the vapor equation of state. Because the vaporization and condensation models described in Sec. III.D are not equilibrium models, the pressure is for the vapor alone rather than for the vapor in equilibrium with its liquid. Thus, the vapor pressure equation [Eq. (III-34)] is not evaluated for the vapor pressure using the liquid temperature; instead, each vapor material is assumed to obey a modified polytropic gas relationship with pressure given as

$$p_{Gm} = \rho_{Gm} R_m T_G , \quad m = 1, \dots, \text{NMAT} , \quad (\text{III-53})$$

where

p_{Gm} is the partial pressure of vapor material m ,

ρ_{Gm} is the microscopic density of vapor material m, and

R_M is a variable gas "constant" for vapor material M.

The most important parameter in this equation is R_M . The algorithm chosen to evaluate R_M is based on the observation that, in the analysis of LMFBR HCDA situations, high vapor densities and significant deviations from ideal behavior generally are calculated near saturated conditions. For example, it would be very difficult to obtain highly superheated sodium vapor except at low density and pressure. However, SIMMER-II has been used with vapor temperatures well above the critical temperature in some cases. For example, in the expansion phase of hypothesized steam explosions, large amounts of fuel can exist at 3000 K, whereas the water critical temperature is 647.3 K. Therefore, variations in R_M are assumed to be based on saturation conditions with a correction for vapor temperatures above the critical temperature. For T_G below $T_{Crt,M}$, R_M is defined by a function f_M , where f_M is a simple empirical fit based on the product of the vapor density and temperature. Also, each component is assumed to be fit independently using saturated vapor conditions. For T_G above $T_{Crt,M}$, the R_M is assumed to approach the infinitely dilute value monotonically. The formulation chosen may be summarized by

$$\ln R_M = g_M \ln f_M + (1 - g_M) \ln R_{Mi} \quad , \quad (III-54)$$

where

$$\begin{aligned} g_M &= 1 \quad , & T_G &\leq T_{Crt,M} \quad , \\ g_M &= a_{G,M} / (T_G - T_{Crt,M} + a_{G,M}) \quad , & T_G &> T_{Crt,M} \quad , & (III-55) \\ f_M &= R_{Mi} \quad , & \rho_{Gm} T_G &< ROGCUT \quad , \\ \ln f_M &= a_{M10} + a_{M20} \ln(\rho_{Gm} T_G) + a_{M30} (\rho_{Gm} T_G)^{1/2} \quad , & ROGCUT &\leq \rho_{Gm} T_G \leq (\rho_{Gm} T_G)_o \quad , \\ \ln f_M &= a_{M11} + a_{M21} (\rho_{Gm} T_G)^{1/2} + a_{M31} \rho_{Gm} T_G \quad , & (\rho_{Gm} T_G)_o &< \rho_{Gm} T_G \leq (\rho_{Gm} T_G)_{Crt} \quad , \\ f_M &= R_{Crt,M} \quad , & \rho_{Gm} T_G &> (\rho_{Gm} T_G)_{Crt} \quad , \end{aligned}$$

with

$$R_{Mi} = (\gamma_M - 1)c_{vGM} ,$$

$$R_{Crt,M} = \frac{P_{Crt,M}}{(\rho_{Crt,M} T_{Crt,M})} ,$$

$$(\rho_{Gm} T_G)_{Crt} = \rho_{Crt,M} T_{Crt,M} ,$$

$$(\rho_{Gm} T_G)_o = \text{the product of the vapor density and temperature at } T_{Sat,M} = 0.95 T_{Crt,M} ,$$

$$\gamma_M = \frac{c_{pGM}}{c_{vGM}} = \text{the ratio of vapor specific heats at infinite dilution for material M,}$$

$$P_{Crt,M} = \text{the critical pressure for material M,}$$

$$\rho_{Crt,M} = \text{the critical density for material M,}$$

$$ROGCUT \text{ and } a_{G,M} = \text{SIMMER input variables, and}$$

$$a_{M10}, a_{M20}, a_{M30}, a_{M11}, a_{M21}, \text{ and } a_{M31} \text{ as fitting constants.}$$

The six fitting constants are determined along the saturation curve with $T_G \leq T_{Crt,M}$ from the following conditions.

- (1) For continuity, $R_M = R_{Mi}$ at $\rho_{Gm} T_G = ROGCUT$.
- (2) At infinite dilution, R_M should be constant; therefore,

$$\frac{d(R_M)}{d(\rho_{Gm} T_G)} = 0 \text{ at } \rho_{Gm} T_G = ROGCUT .$$

- (3) At $T_{Sat,M} = 0.95 T_{Crt,M}$, $R_M = R_{Mo}$ where R_{Mo} is defined using the vapor pressure relationship and $(\rho_{Gm} T_G)_o$.
- (4) Values for R_M must be continuous at $(\rho_{Gm} T_G)_o$.
- (5) For continuity, $R_M = R_{Crt,M}$ at $(\rho_{Gm} T_G)_{Crt}$.
- (6) When a component's partial pressure equals the critical pressure, infinite compressibility is assumed; therefore,

$$\frac{\partial(\rho_{Gm} R_M T_G)}{\partial \rho_{Gm}} = 0 \text{ when } \rho_{Gm} T_G = (\rho_{Gm} T_G)_{Crt} .$$

Again, the one exception to this fitting procedure is the fission-gas component, where R_5 is assumed to be R_{5i} , or

$$R_5 = (\gamma_5 - 1) c_{vG5} \quad (III-56)$$

The total pressure now can be expressed as the sum of the partial pressures

$$P = \sum_{m=1}^{N_{MAT}} P_{Gm} = \sum_{m=1}^{N_{MAT}} \rho_{Gm} R_M T_G \quad (III-57)$$

The gas constant, pressure, and vapor temperature are obtained from Eqs. (III-49) and (III-57) by a Newton-Raphson procedure outlined in Sec. IV.C.2.

c. Single-Phase Flow States. For single-phase flow, the local pressure is obtained from the equation of state for the liquid components. This pressure should be continuous at the transition between single-phase and two-phase flow. For single-material systems, such continuity is obtained by requiring equilibrium conditions between the liquid and the vapor components. Multimaterial systems are more complicated and require additional assumptions to obtain continuity. In SIMMER-II, the materials are assumed to be immiscible; thus, at equilibrium, the vapor pressure for the material mixture is the sum of the vapor pressures for the individual materials. As derived in Appendix B, the resulting pressure for single-phase flow is given by

$$P = \bar{p}_L + \frac{c^2}{1 - \alpha_S} \left(\frac{\alpha_L - \alpha_{Lo}}{\alpha_L} \right) \bar{p}_L \quad (III-58)$$

where

$$\bar{p}_L = \sum_{m=1}^4 \bar{p}_{vM(m)} \quad (III-59)$$

$$\bar{p}_{vM(m)} = \min [p_{vM}(T_{Lm}), p_{T,m}] \quad (III-60)$$

$$P_{T,m} = \min \left\{ \bar{p}_{Lm} \left(\frac{\partial p}{\partial \bar{p}_L} \right)_{k \neq m}, [(\bar{p}_{Gm} + \bar{p}_{Lm}) R_M T_G] / \alpha_G \right\}, \quad (\text{III-61})$$

$$\frac{1}{\epsilon^2} = \sum_{m=1}^{NCLE} \frac{\alpha_{Lm}}{\alpha_L c_{Lm}^2}, \quad (\text{III-62})$$

$$\left(\frac{\partial p}{\partial \bar{p}_L} \right)_{k \neq m} = \left[\frac{1 - \alpha_S}{\epsilon^2} - \frac{\alpha_{Lm}}{\alpha_L c_{Lm}^2} (1 - \alpha_S) \right]^{-1} = \frac{1}{\sum_{\substack{k=1 \\ k \neq m}}^{NCLE} \frac{\alpha_{Lk}}{\alpha_L c_{Lk}^2} (1 - \alpha_S)}, \quad (\text{III-63})$$

and where

- α_{Lo} is the liquid-field volume fraction at the assumed transition point between single- and two-phase flow,
- ϵ^2 is the square of the effective sonic velocity for the liquid-field mixture,
- $\bar{p}_L$ is the sum of the liquid-field vapor pressures,
- $\bar{p}_{vM(m)}$ is the effective liquid vapor pressure for liquid-field energy component m,
- $p_{vM}(T_{Lm})$ is the liquid vapor pressure [Eq. (III-34)] for liquid-field energy component m evaluated at the component temperature and
- c_{Lm} is the liquid sonic velocity for component m.

With the decrease in liquid fuel density as a function of temperature, liquid single-phase pressures sometimes can develop to limit the energy production in a fast reactor disassembly calculation with SIMMER-II. These pressures may be unrealistically high if a constant fuel sonic velocity is used. Consequently, the actual decrease in fuel sonic velocity is programmed using the algorithm

$$c_{L1} = \max [A (1 - T_{L1}/T_{Crt,1})B, C] \quad , \quad T_{L1} < T_{Crt,1} \quad , \quad (III-64)$$

$$c_{L1} = C \quad , \quad T_{L1} \geq T_{Crt,1} \quad ,$$

where A, B, and C are input constants; T_{L1} is the liquid fuel temperature; $T_{Crt,1}$ is the liquid fuel critical temperature; and c_{L1} is the liquid fuel sonic velocity.

A continuous thermodynamic state at the transition from single- to two-phase flow imposes further conditions on the vapor state at the transition. These conditions are relatively trivial for single-component systems; that is, the liquid and vapor components have the same pressure and temperature. For the analytic equation of state, the single-component conditions are expressed as

$$T_G = T_L \quad , \quad \text{and} \quad (III-65)$$

$$p_G = \rho_G R T_G = p_L = p_v(T_L) \quad . \quad (III-66)$$

For multicomponent systems, the situation is somewhat different. In SIMMER-II, the state of each vapor material is assumed to be independent of other vapor materials, and each vapor material is assumed to be in saturated equilibrium with its liquid counterpart. This implies that the vapor internal energy should be defined by

$$e_G = \sum_{m=1}^{N_{MAT}} (\rho_{Gm} e_{vap,M}) \left(\sum_{m=1}^{N_{MAT}} \rho_{Gm} \right)^{-1} \quad . \quad (III-67)$$

For the analytic equation of state, Eq. (III-67), Eq. (III-49), and the definition of $e_{G,M}^*$ are used to eliminate $e_{vap,M}$ and give T_G as

$$T_G = \sum_{m=1}^{N_{MAT}} \rho_{Gm} c_{vGM} T_{Sat,M} \left(\sum_{m=1}^{N_{MAT}} \rho_{Gm} c_{vGM} \right)^{-1} \quad . \quad (III-68)$$

In Eq. (III-68), ρ_{Gm} and $T_{Sat,M}$ remain arbitrary. The condition of saturated equilibrium suggests that the saturation temperatures equal the appropriate liquid temperatures. The vapor density is chosen by assuming

$$\hat{p}_{vM(m)} = p_{Gm} = \rho_{Gm} R_M T_G \quad . \quad (III-69)$$

When T_G is below $T_{Crt,M}$, R_M is only a function of the product $\rho_{Gm} T_G$, and R_M can be calculated iteratively from Eq. (III-69) for known values of $\hat{p}_{vM(m)}$. If an estimated value of T_G is above $T_{Crt,M}$, the iteration for R_M involves g_M from Eq. (III-55). In the present version of SIMMER-II, g_M is estimated using a T_G calculated from Eq. (III-49) using the vapor densities and internal energy from the previous cycle. After R_M is obtained, Eq. (III-69) is solved for ρ_{Gm} , which is substituted into Eq. (III-68) to obtain T_G from known quantities as

$$T_G = \frac{\sum_{m=1}^{NMAT-1} \left[\frac{\hat{p}_{vM(m)} c_{vGM} T_{Lm}}{R_M} \right]}{\sum_{m=1}^{NMAT-1} \left[\frac{\hat{p}_{vM(m)} c_{vGM}}{R_M} \right]} \quad . \quad (III-70)$$

This summation does not include the fission-gas component $m = NMAT$. In single-phase cells, the fission-gas vapor density is assumed to be zero. Further, this result does not guarantee thermal equilibrium conditions at the single-phase/two-phase transition. The vapor-mixture temperature obviously cannot equal all the liquid temperatures simultaneously; hence, temperature differences will exist to drive phase transitions. For a given equation of state and a single vapor temperature, no other recourse is available.

In the previous discussions, the liquid-vapor pressure, p_{vM} , was evaluated at the liquid temperature, T_{Lm} . To account for superheat that delays the single-phase/two-phase transition until the liquid temperature reaches the normal boiling temperature plus the superheat temperature, the saturation temperature and liquid-vapor pressure actually are evaluated at the liquid temperature minus the superheat-temperature increment. When the switch from single- to two-phase occurs, the pressure is continuous. However, because the saturation temperatures are lower than the liquid temperatures from the superheat-temperature

increment, rapid vaporization and pressurization of the two-phase region can occur.

d. Liquid Microscopic Densities. Using constant microscopic densities does not allow the calculation of many single-phase expansion effects that are important in determining the neutronics behavior of reactor transients. However, calculating variable microscopic densities for all components would involve extensive code modifications to be efficient. As a compromise, temperature-dependent liquid microscopic densities are calculated for the first three liquid energy components (fuel, steel, and sodium). The formulas used are

$$\begin{aligned} \rho_{Lm} &= a_{1m} + a_{2m} T_{Lm} + a_{3m} T_{Lm}^2 + a_{4m} T_{Lm}^3, & T_{Melt,M} \leq T_{Lm} \leq \frac{2}{3} T_{Crt,M}, \\ \rho_{Lm} &= \rho_{Crt,M} [1 + b_{1m} \xi_m^{b_{2m}} + b_{3m} \xi_m^2], & \frac{2}{3} T_{Crt,M} < T_{Lm} \leq T_{Crt,M}, \text{ and} \\ \rho_{Lm} &= \rho_{Crt,M} & T_{Lm} > T_{Crt,M}, \end{aligned} \quad (III-71)$$

where

$$\xi_m = 1 - T_{Lm}/T_{Crt,M},$$

$\rho_{Crt,M}$ = the liquid critical density, and

a_{1m} , a_{2m} , a_{3m} , a_{4m} , b_{1m} , b_{2m} , and b_{3m} are input parameters.

2. Tabular Equation-of-State Model. In SIMMER-II, the tabular equation-of-state option retrieves data from the Los Alamos equation-of-state library called SESAME.⁴ This library provides a wider and more realistic data base than can be obtained with an analytic representation.

a. Material Equations of State. SIMMER-II requires equation-of-state tables for reactor fuel, stainless steel, sodium, and B₄C. The fission gas is treated as an ideal gas. The equation-of-state tables are described only generally because (1) the equation-of-state data base changes as better theoretical and experimental results become available and (2) a workable tabular equation-of-state option will require future code modifications. Further, in view of limited implementation, past precedence, and the difficulty in distinguishing between vapor material component quantities requiring a lower case subscript and actual equation-of-state quantities requiring an upper case subscript, all material subscripts will be upper case in describing the tabular equation-of-state

model. Standard subscripting practice will resume in Sec. III.C.3, where vapor mixture properties are described.

Several theoretical models are used in generating equation-of-state tables for the SESAME library. A particularly useful method for reactor safety applications is that of Barnes and DeVault.^{5,6} The treatment is similar to a virial expansion; however, the pressure as a function of density ρ and temperature T is written as

$$p(\rho, T) = p_c(\rho) + RT[\rho + b(T) \rho^2 + c(T) \rho^3] \quad , \quad (\text{III-72})$$

where $p_c(\rho)$ is the 0 K curve, which includes the effects of forces between the atoms in a static configuration. The temperature-dependent terms represent the pressure arising from thermal motion of the atoms. The functions $b(T)$ and $c(T)$ are obtained by matching this expression to a modified Debye model for the crystal at normal density. As shown in Fig. 3, the calculated isotherms have a Van der Waals form below an implied critical temperature. States between points B and C are thermodynamically unstable $[(\partial p / \partial \rho)_T < 0]$ and are not allowed. At equilibrium, a two-phase region is formed in which the pressure is constant between points A and D. The states between points A and B and between points C and D are nonequilibrium states required in the SIMMER-II models for the rates of vaporization and condensation.

Equation-of-state tables for uranium dioxide, iron, sodium, and B_4C are available for use in SIMMER-II. The SESAME tables provide a complete equation of state for the vapor, liquid, and solid states. However, in SIMMER-II, approximations are made to simplify the fluid-dynamics calculations. Hence, the tables are used only to define the vapor equation of state and liquid and vapor states on the coexistence line at this time.

b. Vapor Mixture Rule. For SIMMER-II applications, an efficient algorithm has been developed to compute the equation of state of the vapor mixture from equation-of-state tables for the pure constituents. This mixture rule is constrained to give ideal mixing for ideal gases,⁷ but it gives reasonable results for nonideal gases as well.

A summary of the mixture rule is given in the following. We define the following quantities.

Fig. 3.
Typical pressure isotherm with Van der Waals loop.

x_M is the mass fraction of material M,

ρ is the vapor density,

e is the vapor specific internal energy,

A_M is the molecular weight of material M,

v is the specific volume ($1/\rho$),

p is the pressure,

T is the temperature of the vapor mixture,

c_v is the constant-volume specific heat of the mixture, and

s is the specific entropy of the mixture.

In SIMMER-II, ρ and e are the independent variables and p and T are the dependent variables. The SESAME tables define the following functions.

$p_M(v_M, e_M)$ is the pressure of material M,

$T_M(v_M, e_M)$ is the temperature of material M, and

$e_{cM}(v_M)$, $p_{cM}(v_M)$ is the "cold curve" (pressure and energy at 0 K).

To compute the functions $p(v,e)$, $T(v,e)$ for the mixture, we define

$$\bar{A} = \left(\sum_{M=1}^{N_{MAT}} \frac{x_M}{A_M} \right)^{-1} \quad \text{as the average molecular weight,}$$

$$w_M = \bar{A} \frac{x_M}{A_M} \quad \text{as the mole fraction of material M, and}$$

$$v_M = \bar{A} \frac{v}{A_M} \quad \text{as the specific volume of material M.}$$

The cold curves of the mixture are defined to be

$$e_c(v) = \sum_{M=1}^{N_{MAT}} x_M e_{cM}(v_M) \quad , \quad (III-73)$$

and it follows from the definition of $(de/dv)_s$ that

$$p_c(v) = -\left(\frac{de_c(v)}{dv}\right)_s = \sum_{M=1}^{N_{MAT}} w_M p_{cM}(v_M) \quad . \quad (III-74)$$

An effective energy density for each component can be obtained from the total energy density as

$$e_M = \frac{\bar{A}(e - e_c)}{A_M} + e_{cM} \quad , \quad M = 1, \dots, N_{MAT} \quad . \quad (III-75)$$

Then the temperature and pressure of the mixture are assumed to be given by

$$T(v,e) = \left[\sum_{M=1}^{N_{MAT}} \frac{w_M}{T_M(e_M, v_M)} \right]^{-1} \quad , \quad \text{and} \quad (III-76)$$

$$p(v, e) = \sum_{M=1}^{NMAT} w_M \left[\frac{T}{T_M} p_M(e_M, v_M) + \left(1 - \frac{T}{T_M}\right) p_{cM}(v_M) \right] . \quad (III-77)$$

These equations reduce to ideal mixing when each component is an ideal gas. In addition, this rule [Eqs. (III-76) and (III-77)] is thermodynamically consistent; that is, each material component and the mixture satisfy the general thermodynamic relation

$$\left(\frac{\partial T}{\partial v}\right)_e = \frac{p}{c_v} - T \left(\frac{\partial p}{\partial e}\right)_v . \quad (III-78)$$

In addition to the equation of state for the vapor, SIMMER-II requires a saturation temperature for in the vaporization-condensation model. The "partial pressure" for each material is defined as

$$p_{vM} = w_M \left[\frac{T}{T_M} p_M(v_M, e_M) + \left(1 - \frac{T}{T_M}\right) p_{cM}(v_M) \right] ,$$

$$M = 1, \dots, NMAT . \quad (III-79)$$

The saturation temperature is assumed to be the vapor temperature for the pure component at pressure p_{vM} . Although this definition is correct for a mixture of ideal gases, it is somewhat ambiguous when applied to a mixture of nonideal gases. However, when the liquid and vapor are out of equilibrium, it does induce vaporization or condensation to bring the two phases back to equilibrium.

3. Vapor Mixture Properties. To calculate the exchange functions involving the vapor field, the average thermal conductivity, viscosity, and mass diffusivity of the vapor mixture are necessary. In SIMMER-II, the isolated-component properties are computed by using the Chapman-Enskog kinetic theory of gases⁸ and the Lennard-Jones molecular-potential model.^{9,10}

The thermal conductivity for monoatomic gases is computed as

$$k_{Gm}^{\text{mono}} = 8.314 \cdot 10^{-2} \frac{\left(\frac{T_G}{A_m}\right)^{1/2}}{\sigma_m^2 \Omega_{k,m}}, \frac{W}{m \cdot K}, \quad (\text{III-80})$$

where

A_m is the molecular weight,

σ_m is the characteristic diameter of the molecule ($\text{\AA}$), and

$\Omega_{k,m}$ is the collision integral for material m.

The collision integral is a weak function of temperature and is nearly constant at high temperatures. Its variation with temperature is tabulated in Ref. 11. A least-squares fit was made from this tabulation, resulting in

$$\Omega_{k,m} = 0.92495 + 2.07368 \cdot 10^{-3} \theta_m + 0.719288 \theta_m^{-1.15049} - 5.46452 \cdot 10^{-2} \theta_m^{1/2},$$

where

$$\theta_m = T_G / \epsilon_m \quad \text{and}$$

ϵ_m is an input molecular force constant for material m (K).

For polyatomic gases, the thermal conductivity is given by

$$k_{Gm}^{\text{poly}} = k_{Gm}^{\text{mono}} + 1.32 \left[c_{pGM} - \frac{5}{2} (c_{pGM} - c_{vGM}) \right] \mu_{Gm}, \quad (\text{III-81})$$

where μ_{Gm} is the viscosity of vapor-field material component m.

The viscosity is calculated as

$$\mu_{Gm} = 2.67 \cdot 10^{-6} \frac{(T_G A_m)^{1/2}}{\sigma_m^2 \Omega_{k,m}}, \frac{N \cdot s}{m^2}. \quad (\text{III-82})$$

The mass diffusivity has the form of a binary diffusion coefficient:

$$D_{mk} = 0.018768 \frac{[T_G^3 (\frac{1}{A_m} + \frac{1}{A_k})]^{1/2}}{\sigma_{mk}^2 \Omega_{D,mk} p} , \quad \frac{m^2}{s} , \quad (III-83)$$

where

$\Omega_{D,mk}$ is the collision integral for diffusion, and

p is the local pressure given by Eq. (III-57).

The least-squares fit to the collision integral for diffusion gives

$$\Omega_{D,mk} = 0.82515 + 1.9066 \cdot 10^{-3} \theta_{mk} + 0.66448 \theta_{mk}^{-1.0676} - 5.0246 \cdot 10^{-2} \theta_{mk}^{1/2} ,$$

where $\theta_{mk} = T_G / (\epsilon_m \epsilon_k)^{1/2}$.

The characteristic diameter for the diffusion process is the average for the two materials:

$$\sigma_{mk} = \frac{1}{2} (\sigma_m + \sigma_k) . \quad (III-84)$$

The mixture rules for viscosity and thermal conductivity are calculated as^{9,11}

$$\mu_G = \sum_{m=1}^{NMAT} \left(\frac{w_m \mu_{Gm}}{\sum_{k=1}^{NMAT} w_k \Phi_{mk}} \right) , \quad \text{and} \quad (III-85)$$

$$k_G = \sum_{m=1}^{NMAT} \left(\frac{w_m k_{Gm}}{\sum_{k=1}^{NMAT} w_k \phi_{mk}} \right) , \quad (III-86)$$

$$\text{where } \phi_{mk} = \frac{\left[1 + \left(\frac{\mu_{Gm}}{\mu_{Gk}} \right)^{1/2} \left(\frac{A_k}{A_m} \right)^{1/4} \right]^2}{\sqrt{8} \left[1 + \left(\frac{A_m}{A_k} \right) \right]^{1/2}} . \quad (III-87)$$

D. Exchange-Function Models

The equations of motion are coupled through exchange functions that model mass, momentum, and energy exchange among the three fields and their components. These processes are modeled relatively simply in SIMMER-II, but the models are more complex than those of SIMMER-I. The dispersed-droplet flow regime is assumed for the SIMMER-II exchange-function models, but other flow-regime models can be implemented. However, care must be taken in defining such new models because the logic interrelating the exchange coefficients is tedious and complex and because arbitrary functional forms can cause the numerical procedures to be unstable.

The exchange-function models of SIMMER-II depend on the instantaneous values of the field variables. Because of the pure Eulerian differencing scheme, it currently is not possible to transport configuration histories through the mesh. The use of marker particles might allow some relaxation of this disadvantage. However, a large increase in storage requirements and running time could result. With these caveats, the SIMMER-II exchange-function models can be described as follows.

1. Structure-Configuration Model. The structure-configuration model dynamically tracks the local geometric arrangement of the structure-field components. That arrangement not only includes the intact structural components but also accounts for those components that have melted and then adhered to the structure after refreezing. The model accounts for typical LMFBR subassemblies consisting of cylindrical fuel pins and a subassembly wall material. Thus, each

mesh cell potentially contains both fuel pins and subassembly wall material. As fuel pin material melts and is released to the liquid field, the fuel pin is assumed to retain its cylindrical shape but to decrease its radius uniformly, thus altering its external surface area and the flow hydraulic diameter. If material freezes onto the fuel pin, the pin's radius is assumed to increase uniformly until adjacent fuel pins are predicted to touch. The subassembly wall is assumed to remain as a flat surface until the fuel pins contact. Freezing and melting simply increase and reduce the subassembly wall thickness without changing the wetted surface area.

For fuel pins in a triangular array, contact occurs when the pin volume fraction is approximately 0.9 of the available volume, excluding the volume of subassembly wall. As more material adheres after contact, the exposed surface area to the flow channel decreases. In SIMMER-II, this flow channel is assumed to behave as a cylindrical pipe. Then, as more material adheres to the structure, the radius of the pipe and the wetted surface area are assumed to decrease.

An option in SIMMER-II allows the structure-field control component to be included in a mesh cell for its neutronics effect, but it is decoupled thermally from the other components in the mesh cell. With this option, all materials associated with the control assembly, including the steel and sodium, are modeled by the structure-field control components. The structure-configuration model ignores the presence of the control component, and the calculation proceeds as if only the remaining components occupied volume in the mesh cell. For consistency, the neutronics input for the control material must include the associated steel and sodium. The following discussion assumes that this option is not exercised.

The topology of the structure-field components depends on the initial nature of the intact structure. For example, a mesh cell may contain all the structure-field energy components (fabricated fuel, frozen fuel, cladding, can-wall, and control). (The fission gas components are not pertinent to the structure-configuration model.) If we consider the pin and the components associated with it, the order of the materials from inner radius to outer radius depends on whether the pin was initially a fuel pin or a control pin.

If the pin initially was a fuel pin, the order is assumed to be (1) fabricated fuel, (2) cladding, and (3) frozen control material. Fabricated fuel and cladding are placed first because fuel pins are so fabricated. No frozen fuel is placed as a crust on fuel pins, which agrees with the results of fuel-

freezing tests performed by Spencer.¹² Because there is no component for frozen steel, the steel that freezes on the cladding is not distinguished from cladding, and the cladding is predicted to thicken.

If the pin is initially a control pin, then the order is control material and cladding. Again, no frozen-fuel crust is assumed to be possible on the control pins. Fabricated fuel should not exist because a cell cannot contain both fuel pins and control pins simultaneously except for the option discussed previously. Any control material or cladding that freezes is presumed to thicken the respective component.

The situation in the fission-gas plenum for both fuel and control assemblies also requires special ordering. This sequence is (1) nonflow volume, (2) cladding, and (3) frozen control material. The nonflow volume accounts for the volume inside the cladding that is not available to the flow fields until cladding failure [$\bar{p}_{s5} < (\bar{p}_{s5})_{fail}$].

The layering of the various components on the subassembly wall must account for the presence of frozen control or fuel crusts. If a subassembly wall is associated with a cell containing a fuel pin, the order of layering on the can-wall is frozen fuel before frozen control material. The molten fuel is expected to exist in a fuel-pin channel before the molten control material. If a can-wall is associated with a cell containing a control rod, only frozen fuel is layered on the can-wall because the control material is assumed to exist within the pin.

Two additional cases occur in mesh cells that initially contain can-wall only. If the can-wall is associated with a fuel assembly, then the order of layering on the can-wall is frozen fuel before frozen control material. For a control assembly, the order is frozen control material before frozen fuel. Again, this ordering is based on what component logically might be expected to freeze first.

The structure-configuration model assigns the layering of components in the direction of small to large radius for the pin structure and from the subassembly can-wall toward the flow channel for the can-wall structure. For each component layer, heat-transfer coefficients, surface areas, and hydraulic diameters for the intact structure components are adjusted for melting and freezing. In addition, the total volume fractions of the pin structure and can-wall structure are accumulated to make the appropriate adjustments for the frozen materials. The parameters defining the intact structure (listed in Table VI), are region-dependent and are provided by the user. How these intact

TABLE VI
SIMMER-II REGION PARAMETERS

<u>Position in AAROW</u>	<u>FORTTRAN Variable</u>	<u>Variable</u>	<u>Description (units)</u>
(1)	RGTYP	NTR = RGTYP	<p>Mesh cell indicator for geometric configuration.</p> <p>NTR = 1: Mesh cell is in a fuel pin channel with fuel pellets initially in the mesh cell.</p> <p>NTR = 2: Mesh cell is in a control rod channel with control pellets initially in the mesh cell.</p> <p>NTR = 3: Mesh cell is in a fuel pin channel, but fuel pellets are not present in the mesh cell.</p> <p>NTR = 4: Mesh cell is in a control rod channel, but control pellets are not present in the mesh cell.</p> <p>NTR = 5: Mesh cell is in a fuel pin channel, but no pin structure is present in the mesh cell.</p> <p>NTR = 6: Mesh cell is in a control rod channel, but no pin structure is present in the mesh cell.</p> <p>NTR = 7: No structure exists in the mesh cell.</p> <p>A negative value of RGTYP for a fuel pin channel indicates that the control material component in the mesh cell does not interact thermally with the other components and is used to include its neutronics effect only.</p>
(2)	ALNOFL	α_{nf}	Volume fraction not available for flow.
(3)	PNOFL	P_{nf}	Pressure in the volume fraction not available for flow (Pa).

TABLE VI (CONT)

<u>Position in AAROW</u>	<u>FORTTRAN Variable</u>	<u>Variable</u>	<u>Description (units)</u>
(4)	RPELL	r_{pell}^i	Outer radius of the fuel pellet (m).
(5)	RCLAD	r_{clad}^i	Outer radius of the fuel pin cladding (m). Not used in the current version of SIMMER-II.
(6)	THCANW	t_{can}^i	Thickness of the subassembly can-wall (m).
(7)	APELL	A_{pell}^i	Fuel or control pellet surface area per mesh cell unit volume (m^{-1}).
(8)	ACLAD	A_{clad}^i	Fuel pin cladding surface area per mesh cell unit volume (m^{-1}).
(9)	ACAN	A_{can}^i	Can-wall surface area per mesh cell unit volume (m^{-1}).
(10)	VFPELL	α_{pell}^i	Initial volume fraction of the fuel pellet.
(11)	VFCLAD	α_{clad}^i	Initial volume fraction of the cladding.
(12)	VFCANW	α_{can}^i	Initial volume fraction of the subassembly can-wall.
(13)	DHINGE	D_{hc}^i	Hydraulic diameter for intact geometry (m).
(14)	DHNCLD	D_{hf}^i	Hydraulic diameter for fuel or control pellets with no cladding (m).
(15)	DHNPEL	D_{hcan}^i	Hydraulic diameter for a subassembly with no fuel pins (m).
(16)	HTCPEL	h_{pell}^i	Heat-transfer coefficient for the fuel pellet ($\text{W}/\text{m}^2 \cdot \text{K}$).
(17)	HTOCLD	h_{clad}^i	Heat-transfer coefficient for the cladding ($\text{W}/\text{m}^2 \cdot \text{K}$).
(18)	HTOCAN	h_{can}^i	Heat-transfer coefficient for the subassembly can-wall ($\text{W}/\text{m}^2 \cdot \text{K}$).

TABLE VI (CONT)

<u>Position in AAROW</u>	<u>FORTTRAN Variable</u>	<u>Variable</u>	<u>Description (units)</u>
(19)	TSSF	T_{S1}^i	Minimum fuel temperature for intragranular fission-gas release (K).
(20)	FURSF	f_{ursf}	Fraction of the fabricated fuel that is unrestructured.
(21)	DPURSF	$\Delta\rho_{ursf}$	Pressure difference across the unrestructured fuel annulus (Pa).
(22)	RGRAIN	r_{gr}	Fuel grain radius in the unrestructured fuel (m).
(23)	PERMF	K	Unrestructured fuel permeability (m^2) in input. Later FSC7 stored here. See the input description.
(24)	RPMAX	$r_{p,max}$	Maximum allowed liquid droplet radius (m).
(25)	RPMIN	$r_{p,min}$	Minimum allowed liquid droplet radius (m). Does not apply to the hydraulic diameter limitation.
(26)	RHOSAT	$\rho_{fg,sat}$	Saturated value of fission gas on grain boundaries (atoms/ m^2). Modified to be the actual density (kg/m^3) when the code is executing.
(27)	ALNFCW	$\alpha_{nf,cw}$	Volume not available for flow when radial motion at the right-hand boundary of a cell is prohibited.
(28)	GPFLG	F_{gp}	Parameter used in modeling to simulate fuel flow in intersubassembly gaps.

variables are adjusted to account for freezing and melting of the structure-field components is discussed below.

The model will be described in terms of how the pin volume and the can-wall volume are constructed. To obtain a pin contribution, either fabricated fuel, cladding, or control material must exist and NTR must be less than 5. If NTR is less than 5 but no pin structure is found in the mesh cell, NTR is reset to 5 or 6 as appropriate. (If no structure is found to exist, NTR is reset to 7, and the structure configuration model is bypassed.)

If $NTR = 1$ or 2 and both fabricated fuel/control and cladding exist, the cladding is assumed to contain within it the initial volume of the pellet. Therefore, the input initial pellet volume fraction α_{pell}^i is used for the transient pellet volume fraction α_{pell} . In this situation, the pellet surface area per unit volume is not modified from the input surface area A_{pell}^i , and the input heat-transfer coefficient h_{pell}^i for the pellet is used for the transient conditions. The first contribution to the pin volume fraction, α_p , is given by

$$\alpha_p = \alpha_{pell}^i + \alpha_{nf} \quad . \quad (III-88)$$

If $NTR = 1$ or 2 and no cladding exists, the transient fabricated-fuel volume fraction α_{S1} , or the transient control-material volume fraction α_{SS} (depending on the type of mesh cell) is used for the transient pellet volume fraction. The change in the pellet volume fraction is assumed to result in a uniform change in the pellet radius. Because the pellet volume is proportional to the square of the pellet radius and the pellet surface area is proportional to the pellet radius, the transient pellet surface area is given by

$$A_{pell} = A_{pell}^i \left(\frac{\alpha_{pell}}{\alpha_{pell}^i} \right)^{1/2} \quad . \quad (III-89)$$

Similarly, the pellet heat-transfer coefficient is proportional to the pellet radius, and the transient value is obtained by

$$h_{pell} = h_{pell}^i \left(\frac{\alpha_{pell}}{\alpha_{pell}^i} \right)^{1/2} \quad . \quad (III-90)$$

The transient pellet volume fraction is then the first contribution to the pin volume fraction:

$$\alpha_p = \alpha_{pell} \quad . \quad (III-91)$$

If $NTR = 1$ or 2 and no pellets exist, cladding must exist. (Otherwise NTR would have been reset.) Present SIMMER-II coding treats these cases identically to the cases with $NTR = 3$ or 4 . The exception is that control crusts are prohibited if $NTR = 2$; otherwise they are permitted. For all these cases, the pellet area and heat transfer coefficient are ignored, and the first contribution to the pin structure volume is given by

$$\alpha_p = \alpha_{pell}^i + \alpha_{nf} \quad , \quad (III-92)$$

where the pellet volume fraction must be input as zero if it is not to contribute to α_p with $NTR = 3$ or 4 .

The next component to be considered is cladding. Assuming that $NTR < 5$, its transient volume fraction is

$$\alpha_{clad} = \alpha_{s5} \quad . \quad (III-93)$$

The input initial cladding volume fraction, α_{clad}^i is compared with α_{clad} after adding α_p (obtained above) to the pin volume fraction. Again, any differences are assumed to result from a uniform change in the radius of the pin. Therefore, the transient cladding outer surface area per unit volume is

$$A_{clad} = A_{clad}^i \left(\frac{\alpha_{clad} + \alpha_p}{\alpha_{clad}^i + \alpha_p} \right)^{1/2} \quad . \quad (III-94)$$

The cladding inner-surface area per unit volume is the pellet surface area. To obtain the transient cladding heat-transfer coefficient h_{clad} , the cladding thickness is assumed to be proportional to its volume. Hence, the transient heat-transfer coefficient is obtained by

$$h_{clad} = h_{clad}^i \left(\frac{\alpha_{clad}^i}{\alpha_{clad}} \right) \quad , \quad (III-95)$$

where h_{clad}^i is the input cladding heat-transfer coefficient. The pin-structure volume fraction is incremented by the cladding volume fraction, or

$$\alpha_p = \alpha_{pell}^i + \alpha_{nf} + \alpha_{clad} \quad . \quad (III-96)$$

The can-wall surface area initially is assumed to remain constant as the can-wall thickness decreases or increases. Thus, the input initial can-wall surface area per unit volume A_{can}^i is used for the transient value. In addition, the decrease or increase of the can-wall thickness changes the can-wall heat-transfer coefficient linearly such that

$$h_{can} = h_{can}^i \frac{\alpha_{can}^i}{\alpha_{can}} \quad , \quad (III-97)$$

where $\alpha_{can} = \alpha_{s6}$ is the transient can-wall volume fraction.

Before the frozen materials, frozen fuel, and control are considered, the change in fluid flow area and hydraulic diameter that result from α_{pell} , α_{clad} , and α_{can} differing from α_{pell}^i , α_{clad}^i , and α_{can}^i are calculated. The reference fluid flow area per unit volume, A_{flow}^i , is calculated from

$$A_{flow}^i = \frac{A_{Tot}^i D_h^i}{4 \Delta z_j} \quad , \quad (III-98)$$

where

$$A_{Tot}^i = A_{pin}^i + A_{can}^i,$$

$$A_{pin}^i = A_{clad}^i \text{ if cladding exists,}$$

$$A_{pin}^i = A_{pell}^i \text{ if no cladding exists but pellets do exist,}$$

$$A_{pin}^i = 0 \text{ if no pin exists,}$$

$D_h^i = D_{hc}^i$ if cladding exists,

$D_h^i = D_{hf}^i$ if no cladding exists but pellets do exist,

$D_h^i = D_{hcan}^i$ if no pin exists, and

Δz_j is the axial mesh spacing for the cell in question.

This formula can be related to the standard definition of the hydraulic diameter by noting that $A_{Tot}^i/\Delta z_j$ is the wetted perimeter per unit volume for surfaces in a cell. Now A_{flow}^i is adjusted by using the formula

$$A_{flow} = A_{flow}^i - [(\alpha_{can} - \alpha_{can}^i) + (\alpha_p - \alpha_p^i)]/\Delta z_j, \quad (III-99)$$

where

$\alpha_p^i = 0$ if no pin exists,

$\alpha_p^i = \alpha_{pell}^i$ if pellets exist but no cladding exists, and

$\alpha_p^i = \alpha_{pell}^i + \alpha_{nf} + \alpha_{clad}^i$ if cladding exists.

Depending on whether $\alpha_p > 0.9 (1 - \alpha_{can})$, two paths are possible. Where this inequality is not satisfied, contact of fuel pins is not assumed and

$$A_{Tot} = A_{pin} + A_{can}^i, \quad (III-100)$$

where

$A_{pin} = A_{clad}$ if cladding exists (Eq. III-94),

$A_{pin} = A_{pell}$ if no cladding exists but pellets exist (Eq. III-89), and

$A_{pin} = 0$ if no pin exists.

In this case, the hydraulic diameter is redefined by

$$D_h = \frac{4\Delta z_j A_{flow}}{A_{Tot}} \quad (III-101)$$

If the previous inequality is satisfied, pins are assumed to touch, and both surface areas reduced by an identical multiple so that

$$A_{Tot} = \left(\frac{A_{flow}}{A_{flow}^i} \right)^{1/2} A_{Tot}^i \quad (III-102)$$

This means the hydraulic diameter is reduced by

$$D_h = \left(\frac{A_{flow}}{A_{flow}^i} \right)^{1/2} D_h^i \quad (III-103)$$

If Eq. (III-102) is used, the initial values for A_{clad} and A_{can} now are replaced, but the heat-transfer coefficients do not change.

The frozen fuel and control represent the final adjustments to the structure configuration. These materials have no initial conditions that can be adjusted for the transient. Furthermore, although frozen fuel is assumed to exist only on can-walls, control material can be frozen either on the pin structure or on the can-wall structure. To simplify calculations in SIMMER-II, this control material is assumed to freeze in layers of equal thickness on the pin structure and on the can-wall structure. The detailed treatment is as follows.

First, the volume fraction of a frozen component is determined. All frozen fuel is assigned to can-wall structure. Control material is divided between the pin structure and the can-wall structure in proportion to their surface areas. These surface areas for heat transfer are the previously established values for A_{pin} and A_{can} .

Second, the new cross-sectional area of the flow channel is computed as

$$A_{flow} = A'_{flow} - \alpha_{froz}/\Delta z_j \quad (III-104)$$

where A'_{flow} is the flow channel cross-sectional area without the frozen material, and α_{froz} is the volume fraction of frozen material. For the case of frozen control material the pin volume fraction changes. If $\alpha_p < 0.9 (1 - \alpha_{can})$, $A_{can} = A_{can}^i$, and control material can modify the pin external surface by

$$A_{pin} = A'_{pin} \left\{ \frac{\alpha'_p + [\alpha_{froz} A'_{pin} / (A'_{pin} + A_{can}^i)]}{\alpha'_p} \right\}^{1/2}, \quad (III-105)$$

where

A_{pin} is the pin surface area per unit volume when the frozen material is included,

A'_{pin} is the pin surface area per unit volume when the frozen material is not included,

α'_p is the pin volume fraction when the frozen material is not included, and

α_{can} is the can volume fraction when the frozen material is included.

For all cases with frozen material, if $\alpha_p > 0.9 (1 - \alpha_{can})$, A_{pin} and A_{can} are adjusted by

$$A_{pin} = A'_{pin} \left(\frac{A_{flow}}{A'_{flow}} \right)^{1/2}, \quad \text{and} \quad (III-106)$$

$$A_{can} = A'_{can} \left(\frac{A_{flow}}{A'_{flow}} \right)^{1/2}, \quad (III-107)$$

where A'_{can} is the can-wall surface area per unit volume when frozen material is not included.

The channel hydraulic diameter now can be redefined by Eq. (III-101) if $\alpha_p < 0.9 (1 - \alpha_{can})$ or by

$$D_h = D_h' \left(\frac{A_{Flow}}{A_{Flow}'} \right)^{1/2}, \quad \text{if } \alpha_p > 0.9 (1 - \alpha_{can}), \quad (III-108)$$

where D_h' is the hydraulic diameter without the frozen material. The total surface area finally is recalculated similarly to Eq. (III-100) or Eq. (III-102).

Third, heat-transfer coefficients are defined based on a quasi-static approximation using the crust thickness. The frozen control-material thickness is given by

$$t_{froz} = \frac{2 \alpha_{froz}}{A_{tot} + A_{tot}'} , \quad (III-109)$$

where A_{tot} is the total surface area per unit volume for the pin and can-wall structure after the frozen material is included, and A_{tot}' is the total surface area per unit volume for the pin and can-wall structure before the frozen material is included. The frozen fuel-material thickness is given by

$$t_{froz} = \frac{\alpha_{froz}}{A_{can}} . \quad (III-110)$$

The heat-transfer coefficient for the frozen material is then

$$h_{froz} = \frac{2k_{froz}}{t_{froz}} , \quad (III-111)$$

where k_{froz} is the thermal conductivity of the frozen material.

This calculative sequence is done twice, with control or fuel treated first, depending on the required order (from the value of NTR). When the sequence is finished, the structure-configuration model has determined the heat-transfer paths among the components, the contact surface areas among the components, the heat-transfer coefficient for each component, and the hydraulic

diameter for the flow channel. In addition, the structure-configuration model has identified the components that form the exterior of the pin structure and the can-wall structure and their surface areas and heat-transfer coefficients. It is these two structure components that interact thermally with the liquid- and vapor-field components in the flow channel. A further delineation of the effects of various NTR values will be given in Sec. IV.C.

2. Can-Wall Failure Model. The SIMMER can-wall failure model restrains the r or x direction fluid motion at mesh cell interfaces prescribed to have an intact can-wall. Fluid motion across such an interface is allowed only after the model predicts the occurrence of can-wall failure. SIMMER-II bases that failure on (a) the molten state of the structure-field can-wall component and (b) a simple pressure-dependent failure criterion.

Both can-walls that separate two subassemblies are modeled by assuming the mesh cell interface lies between the two can-walls and assigning each can-wall to the mesh cell where it resides. The two failure criteria are as follows.

(a) Failure at the right-hand boundary of the i th cell occurs if

$$(\bar{\rho}_{s6})_i < (F_{gp})_i * ROSFAL \quad , \quad \text{and}$$

$$(\bar{\rho}_{s6})_{i+1} < (F_{gp})_{i+1} * ROSFAL \quad , \quad (III-112)$$

where ROSFAL is an input constant, F_{gp} is region parameter 28 from Table VI, and the j th subscript is ignored for convenience. This means that densities of both adjacent can-walls must fall below prescribed values before failure is allowed.

(b) Failure at the right-hand boundary of the i th cell also occurs if

$$p_i - p_{i+1} \geq (p_y)_i \quad \text{and} \quad (\bar{\rho}_{s5})_{i+1} \leq RSCLAD \quad , \quad \text{or}$$

$$p_{i+1} - p_i \geq (p_y)_{i+1} \quad \text{and} \quad (\bar{\rho}_{s5})_i \leq RSCLAD \quad , \quad (III-113)$$

where

p_i is the pressure in the i^{th} cell ,

RSCLAD is an input parameter, and

$$p_y = \left[\frac{\alpha_{\text{can}}}{\alpha_{\text{can}}^i} \right]^2 t_{\text{can}}^i \sigma_y(T_{S4}) / D_{\text{hcan}}^i \quad , \quad (\text{III-114})$$

= yield pressure because of circumferential stress,

where t_{can}^i is the can-wall thickness, $\sigma_y(T_{S4})$ is an input yield curve based on the can-wall temperature (T_{S4}), and the j^{th} subscript is ignored for convenience. This criterion says that failure will occur if the circumferential yield stress is exceeded and cladding does not impede motion in the direction of failure. As currently implemented in SIMMER-II, both adjacent cells must be two-phase for this criterion to be satisfied because small motions are generally sufficient to relieve single-phase pressures.

Unless high pressures do develop, generally two adjacent can-walls must at least get close to the steel melting temperature before failure can occur. This has led to an algorithm to transfer heat to an adjacent can-wall when the original can-wall in a cell has vanished. This algorithm is known as the "can-wall set-over model." In this model, the can-wall on left of a cell, the can-wall on the right of a cell, or can-walls on both the left and right of a cell can have their densities put into a cell during execution of the structure configuration model. Definition of a hydraulic diameter, heat-transfer coefficient, and surface area for the temporarily set-over can-wall allows exchange coefficients to be calculated. The initial heat-transfer coefficient is h_{can}^i of the receiving cell. The initial hydraulic diameter and the can-wall area per unit volume depend on whether (r,z) or x-y geometry has been selected. If cylindrical coordinates are selected,

$$A_{\text{can}}^i = 2r_{i-1} / (r_i^2 - r_{i-1}^2) \text{ for a can-wall from the left,}$$

$$A_{\text{can}}^i = 2r_i / (r_i^2 - r_{i-1}^2) \text{ for a can-wall from the right, and} \quad (\text{III-115})$$

$$A_{\text{can}}^i = 2(r_{i-1} + r_i) / (r_i^2 - r_{i-1}^2) \text{ for a can-wall from both the left and right,}$$

where r_i is the outer radius of the i^{th} SIMMER-II ring. Essentially, the outer periphery of a SIMMER-II ring is set into an adjacent cell. In this case, the hydraulic diameter is D_{hcan}^i of the receiving cell. If rectangular geometry is selected,

$$A_{\text{can}}^i = 1/\Delta x_i \quad \text{and}$$

$$D_h^i = 4 \Delta x_i \quad (\text{III-116})$$

for a can-wall transferred from the left or the right, and

$$A_{\text{can}}^i = 2/\Delta x_i \quad \text{and}$$

$$D_h = 2 \Delta x_i \quad (\text{III-117})$$

for a can-wall transferred from both the left and the right, where Δx_i is the width of the i^{th} cell. In this case, the receiving cell models the set-over can-wall as a single plate or a pair of plates. Modifications to the hydraulic diameter, the heat-transfer coefficient, and possibly the surface area occur as discussed in the structure configuration model. In other words, the density and consequently the volume fraction of the set-over can-wall is compared with α_{can}^i input in the region parameters. If can-walls are set-over from both the left and the right, the set-over densities are added and divided by twice the microscopic can-wall density to find the temporary set-over volume fraction.

By using this can-wall set-over model, the high-temperature components of a subassembly can melt the can-walls in both directions. At the same time, an adjacent subassembly with a colder composition can cool its externally heated can-wall. Thus, this model allows intermesh heat transfer in SIMMER-II. When adjacent mesh cell compositions are coupled by heat transfer to remaining can-wall, fuel freezing and melting, that is, crust formation or removal, occur in the receiving cell. Can-wall melting occurs only in the mesh cell in which the remaining can-wall resides, that is, the donor cell. The option to freeze steel on the set-over can-wall has not been programmed. Instead, steel particles are produced. The heat transfer details, for example, how energy is divided between two donating can-walls, are given in Sec. IV.C.6.

This can-wall failure model is used in SIMMER-II unless the user specifies otherwise. Through the variable FAIL3, the user is able to set the can-wall failure test for fluid motion and the intermesh heat-transfer evaluation. The three values that control these features of the model are defined in Table VII.

TABLE VII
FAIL3 VALUES FOR CAN-WALL FAILURE

<u>FAIL3 Value</u>	<u>Test Can-Wall Densities for Failure</u>	<u>Evaluate Intermesh Heat Transfer to Remaining Can Wall</u>
+1	yes	yes
0	no	no
-1	no	yes

In SIMMER-II, the FAIL3(IJ) variable is initialized to the value +1 for each r or x direction mesh cell interface. The user may override this value by redefining a value through IFAIL(I) for all interfaces in the Ith mesh cell column or through FAIL3(IJ) for selected right interfaces of each mesh cell. At each time step during the analysis, SIMMER-II tests each interface with a value of +1 for FAIL3 for proposed failure. If either failure criterion is satisfied, FAIL3 is reset to -1. During the heat-transfer calculation, intermesh heat transfer is implemented only at interfaces flagged by ± 1 values.

Finally, a few words are needed about the use of F_{gp} , which appeared in Eq. (III-112). The primary purpose of this parameter is modeling intersubassembly gap fuel removal when neighboring subassemblies represent a cold blanket ring rather than hot core fuel. The gap modeling then is programmed to disappear when the blanket wall has melted. By giving F_{gp} a large value, for example, $F_{gp} = 5$ in a blanket ring, radial motion into a blanket region can be started while the blanket can-wall still possesses its original density.

To confine flow to the intersubassembly gap channels, interior blanket subassembly flow passages must be modeled by a no-flow volume, and no heat transfer to the pin structure should exist. Assistance on this later point has been provided by setting the pellet and cladding heat-transfer coefficients and surface areas to 10^{-10} if $F_{gp} > 1$.

Nominally, set-over of the blanket can-wall surface into the driver subassembly would occur after radial motion is allowed. Set-over is undesirable in the current situation because both the modeled gap and the driver subassembly would be transferring heat to the blanket can-wall. Consequently, set-over is stopped if F_{gp} is greater than unity. This means that the neighboring rings will not see a wall surface from a ring with $F_{gp} > 1$ nor can a surface be transferred into a region with $F_{gp} > 1$.

When the blanket can-wall melts, we want to model the interior of the sub-assembly, not the gap. This change is made by resetting the region parameter, PAREG(IJ), to unity if $\bar{p}_{s6}$ equals zero. Parameter region 1 is assumed to contain unity for F_{gp} . The original blanket cell no-flow volume is reassigned initially to sodium vapor with a pressure of 2 atm and a temperature of 1209 K. If zero is entered, the default value for F_{gp} is unity.

3. Liquid-Droplet Particle-Size Model. The liquid droplet size determines the liquid-liquid heat-transfer coefficients and areas, the liquid-vapor heat-transfer coefficients and areas, and the momentum-exchange force between the liquid field and the vapor field. In SIMMER-II, the liquid particle sizes are determined from five conditions and phenomena: (1) fluid dynamic breakup based on a Weber number criterion, (2) the hydraulic diameter, (3) the liquid mass in each mesh cell, (4) flashing breakup, and (5) liquid droplet coalescence.

The Weber number criterion assumes that if the inertial forces acting on a droplet exceed the surface tension forces on the droplet, the droplet will break up. The Weber number for a droplet of radius r_{pm} is

$$We_m = \frac{\rho_G |V_{GL}|^2 2r_{pm}}{\sigma_{Lm}}, \quad m = 1, \dots, 4, \quad (III-118)$$

where

We_m is the Weber number for the particle,

ρ_G is the vapor microscopic density,

$|V_{GL}|$ is the absolute velocity difference between the liquid field and the vapor field,

r_{pm} is the droplet radius for liquid-field energy component m, and

σ_{Lm} is the interfacial surface tension between the vapor and liquid-field energy component m.

The critical Weber number for droplets has been found to vary from 1 to 20, depending on the flow situation.¹³ Droplets that lead to a Weber number greater than an input critical value are unstable and are broken up by the algorithm described by Eqs. (IV-47) to (IV-65).

The hydraulic diameter calculated from the structure-configuration model provides a second limitation on the particle radius. Thus, we have

$$r_{pm} = \frac{D_h}{2} , \quad m = 1, \dots, 4 , \quad (\text{III-119})$$

where D_h is the hydraulic diameter of the flow channel. The third criterion is based on the volume fraction of the liquid-field energy components in the mesh cell. A droplet cannot be larger than the total volume of liquid present. The volume of a liquid-field energy component in mesh cell i, j is

$$V_{Lm} = \alpha_{Lm} V_{i,j} , \quad (\text{III-120})$$

where

α_{Lm} is the volume fraction of liquid-field energy component m and

$V_{i,j}$ is the volume of mesh cell i, j .

If the liquid volume represents one single droplet, then its volume is also

$$V_{Lm} = \frac{4\pi}{3} r_{pm}^3 . \quad (\text{III-121})$$

The volume of the smallest mesh cell is used to give the droplet radii. Otherwise, the variable mesh cell spacing will give different results for identical conditions in the mesh cells. Also, to eliminate the effect of the presence of structure on the macroscopic densities, the right side of Eq. (III-120) is divided by $(1 - \alpha_S)$. Thus,

$$r_{pm} = \left[\frac{3}{4\pi} \frac{\alpha_{Lm}}{1 - \alpha_S} \min_{i,j} (V_{i,j}) \right]^{1/3} . \quad (\text{III-122})$$

If the pressure within a droplet exceeds the ambient pressure by more than that supported by the surface tension, the droplet will break up. The pressure within a droplet is assumed to be the component vapor pressure at the droplet temperature. Thus, the fourth estimate of the droplet size is

$$r_{pm} = \frac{2\sigma_{Lm}}{p_{vm}(T_{Lm}) - p} \quad . \quad (III-123)$$

This droplet radius is only evaluated for positive denominators.

The coalescence model, which is described in detail in Appendix C, is based on the notion that discrete packets of liquid will undergo collisions as a result of random velocity fluctuations that may be acceleration, turbulence, or mechanically induced. A collision frequency is defined from an average random velocity magnitude. From this collision frequency, a coalescence rate can be estimated and hence a rate of growth of liquid droplets. This growth is expressed as

$$\frac{\partial r_{pm}}{\partial t} = \text{COAL} \left(\frac{\alpha_{Lm}}{1 - \alpha_S} \right) \quad , \quad (III-124)$$

where

r_{pm} is the droplet radius of liquid-field component m ,

α_{Lm} is the volume fraction of liquid-field component m ,

α_S is the total structure-field volume fraction, and

COAL is an input coefficient that embodies both the coalescence probability per collision and the magnitude of the random velocity (Eq. (C-7) of Appendix C).

The value of COAL is highly uncertain because the magnitude of the random velocity depends on many factors, including the local field velocity, acceleration effects; geometry effects and the probability of coalescence per collision may depend on such quantities as surface tension, velocity of impact, and angle of impact. A COAL value of 1 can be interpreted as a coalescence probability of about unity with a random velocity of 10 m/s. The form of the coalescence equation is reasonable in that the rate of coalescence is proportional to the quan-

tity of liquid in the flow channel. Also, as expected, the coalescence rate tends to go to zero as the population of droplets becomes small.

The uncertainty associated with the selection of a single droplet size to represent a distribution of particles is difficult to assess because the exchange processes depend on droplet size in different ways. For example, liquid-liquid heat transfer is approximately proportional to $1/r_p^2$, turbulent liquid-vapor heat transfer goes as $1/r_p^{1.2}$, turbulent liquid-vapor drag goes as $1/r_p$, laminar liquid-vapor drag goes as $1/r_p^2$, and so on. Therefore, it is impossible to select a single size from a distribution that will produce exchange rates consistent rates with those obtained by integrations over the distribution. A further difficulty arises because the inventories of transported entities tend to be associated with the larger droplets or particles, whereas the transport rates are controlled by the smaller sizes. Thus, if a characteristic size could be selected that preserves the distribution rates, the tendency would be to overestimate the integrated transport over time and to shift artificially the inventories to the smaller size particles.

Characterization of a particle distribution with a single representative particle size is nearly impossible. However, to account in some manner for the expected shift in transport rates to smaller particle sizes, the particle radii determined by the Weber criterion and the flashing breakup model are reduced by a DSDM (droplet size distribution multiplier) factor. The radii in these two models are altered by DSDM because they determine the maximum allowed radius, not the average radius, for the phenomena. Thus, the factor DSDM not only accounts for the distribution effects but also accounts for the ratio of the average radius to the maximum radius from Weber and flashing breakup. The shift in characteristic size produced by DSDM is incorporated in the coalescence model automatically through the radius from the previous time step. Distribution effects on the coalescence rate are not considered.

SIMMER-II often calculates the existence of small amounts of liquid in a mesh cell by artificial diffusion of mass. As a result, r_{pm} can be sufficiently small to cause numerical problems. However, because the small amount of liquid is not important to the total solution of the accident problem, r_{pm} is set to an input minimum value.

In quiescent single-phase liquid, there is no reasonable constraint on the maximum size of liquid droplets if there are no geometric size limitations. To prevent the sizes from becoming unreasonably large, an input maximum radius has been introduced. This maximum radius provides the option of specifying the size

completely by making the input maximum and minimum radii identical. Also, this capability permits the user to simulate expected liquid-liquid heat-transfer rates in mesh cells with multicomponent, single-phase liquid flow by forcing the surface area to represent a physical situation such as the initial distributed character of the liquid components.

The maximum liquid-field droplet radii apply to all liquid-field components at a particular location. They are input as part of the parameter region data. Hence, the maximum radii can be varied over the calculative mesh. However, when transport from one part of the mesh to another occurs, a size change may occur with no physical basis. The algorithm for the droplet size at any particular location is described after Eq. (IV-65).

4. Energy-Exchange Models. Heat transfer between structure-field energy components, between liquid-field energy components, and between liquid-field and structure-field energy components is computed by multiplying the temperature difference between the interacting components by the appropriate heat-transfer coefficient and contact surface area. Heat transfer between the vapor field and the liquid- and structure-field components is assumed to occur only during phase transition, except for a structure-vapor case when the liquid is limited.

a. Structure-Structure Heat Transfer. The heat-transfer coefficients and the heat-transfer paths for each structure component were determined in the structure-configuration model. The structure-structure heat transfer then is calculated from the heat-transfer coefficients and areas as

$$q_{HSmSk} = \frac{h_{Sm} h_{Sk} A_{Smk}}{h_{Sm} + h_{Sk}} (T_{Sk} - T_{Sm}) \quad , \quad (III-125)$$

where

q_{HSmSk} is the heat-transfer rate per unit volume from structure-field energy component k to structure-field energy component m ,

h_{Sm} is the heat-transfer coefficient for structure-field energy component m ,

h_{Sk} is the heat-transfer coefficient for structure-field energy component k ,

A_{Smk} is the heat flow area per unit volume between structure-field energy components m and k ,

T_{Sm} is the temperature of structure-field energy component m, and

T_{sk} is the temperature of structure-field energy component k.

b. Liquid-Structure Heat Transfer. As stated previously, the structure-side heat-transfer coefficient has been determined in the structure configuration model. The surface areas exposed to the liquid and vapor fields are the final pin and can-wall surface areas, A_p and A_{can} , respectively.

The liquid-side heat-transfer coefficient is calculated using a generalized single-phase convective correlation of the form^{14,15}

$$Nu_{Lm} = C_{LS1m} Re_{Lm}^{C_{LS2m}} Pr_{Lm}^{C_{LS3m}} + C_{LS4m} \quad , \quad m = 1, 2, 3, 4, \quad (III-126)$$

where

$$Nu_{Lm} = \frac{h_{LSm} D_h}{k_{Lm}} \quad \text{is the Nusselt number for liquid-field energy component m,}$$

$$Re_{Lm} = \frac{\rho_{Lm} |\vec{V}_L| D_h}{\mu_{Lm}} \quad \text{is the Reynolds number for liquid-field energy component m,}$$

$$Pr_{Lm} = \frac{c_{vLM} \mu_{Lm}}{k_{Lm}} \quad \text{is the Prandtl number for liquid-field energy component m,}$$

C_{LS1m} , C_{LS2m} , C_{LS3m} , C_{LS4m} are input coefficients for the correlation for liquid-field energy component m,

h_{LSm} is the heat-transfer coefficient of liquid-field energy component m,

k_{Lm} is the thermal conductivity of liquid-field energy component m, and

μ_{Lm} is the viscosity of liquid-field energy component m.

The heat flow to liquid energy component m from the surface of structure component k is given by

$$q_{ILmSk} = A_{LmSk} h_{LSm} (T_i - T_{Lm}) \quad , \quad (III-127)$$

where

T_i is the structure surface temperature and

A_{LmSk} is the structure area seen by liquid component m.

The heat flow from the surface to the interior of structure component k is given by

$$q_{ISkL} = A_{Sk} h_{Sk} (T_i - T_{Sk}) \quad , \quad (III-128)$$

where A_{Sk} is the total area of structure component k exposed to fluid. The area seen by liquid component m, A_{LmSk} , is A_{Sk} times a contact coefficient, Λ_m ,

where

$$\Lambda_m = \alpha_{Lm} / (1 - \alpha_S) \quad \text{if} \quad \alpha_L \leq 0.025 \quad , \quad \text{and}$$

$$\Lambda_m = \frac{\alpha_{Lm}}{\alpha_L (1 - \alpha_S)} [\alpha_L \min (2R, 1) + \alpha_G \max (0, 2R-1)] \quad \text{if} \quad \alpha_L > 0.025 \quad ,$$

(III-129)

where R is an input variable ($0 \leq R \leq 1$). A value of zero for R gives no contact of the fluid with the wall. A value of unity for R simulates a film contact. A value of one-half for R assumes that the contact area is proportional to the liquid-field volume fraction. A similar relationship is used later for momentum transfer.

There are two cases for the structure surface temperature. Case 1 considers either a pin surface with fabricated fuel or a can-wall surface with a crust. Case 2 considers either a pin surface with cladding or a can-wall surface with no crust. In either case we solve for an implied T_i by summing all the energy flows from the surface to zero, or

$$\sum_m q_{ILmSk} + q_{ISkL} = 0 \quad . \quad (III-130)$$

Then, the details of the two cases are as follows.

The liquid-liquid heat-transfer model is based on collisions between droplets in a manner similar to the droplet coalescence model. The total heat-transfer rate is the product of the collision frequency and the energy exchange per collision. A detailed derivation of the model is presented in Appendix D.

The resulting expression for the heat-transfer rate per unit volume is

$$q_{HLmLk} = R_{mk} \frac{45}{8} \frac{1}{\frac{r_{pm}}{k_{Lm}} + \frac{r_{pk}}{k_{Lk}}} \frac{\alpha_{Lm} \alpha_{Lk}}{(1 - \alpha_S)} \frac{r_{pm}^2 + r_{pk}^2}{r_{pm}^3 r_{pk}^3} (r_{pm} + r_{pk}) r_c^2 (T_{Lk} - T_{Lm}) ,$$

(III-132)

where

$$r_c = r_{pm} \text{ if } r_{pm} \leq r_{pk} ,$$

$$r_c = r_{pk} \text{ if } r_{pm} > r_{pk} , \text{ and}$$

$$R_{mk} = \text{constant} .$$

The constant R_{mk} represents the uncertainties in the contact area per collision and contact time.

In situations where either r_{pm} or r_{pk} becomes very small, Eq. (III-132) produces heat-transfer rates larger than those that can occur physically. The capability of the larger droplet to reject or receive heat is limited regardless of the heat-transfer capability of the smaller droplet. These limits are

$$(q_{HLmLk})_{\max} = 15 R_{mk} \alpha_{Lk} \frac{k_{Lk}}{r_{pk}^2} (T_{Lk} - T_{Lm}) ,$$

(III-133)

for small r_{pm} and

$$(q_{HLmLk})_{\max} = 15 R_{mk} \alpha_{Lm} \frac{k_{Lm}}{r_{pm}^2} (T_{Lk} - T_{Lm}) ,$$

(III-134)

The liquid-liquid heat-transfer model is based on collisions between droplets in a manner similar to the droplet coalescence model. The total heat-transfer rate is the product of the collision frequency and the energy exchange per collision. A detailed derivation of the model is presented in Appendix D.

The resulting expression for the heat-transfer rate per unit volume is

$$q_{HLmLk} = R_{mk} \frac{45}{8} \frac{1}{\frac{r_{pm}}{k_{Lm}} + \frac{r_{pk}}{k_{Lk}}} \frac{\alpha_{Lm} \alpha_{Lk}}{(1 - \alpha_S)} \frac{r_{pm}^2 + r_{pk}^2}{r_{pm}^3 r_{pk}^3} (r_{pm} + r_{pk}) r_c^2 (T_{Lk} - T_{Lm}) ,$$

(III-132)

where

$$r_c = r_{pm} \text{ if } r_{pm} \leq r_{pk} ,$$

$$r_c = r_{pk} \text{ if } r_{pm} > r_{pk} , \text{ and}$$

$$R_{mk} = \text{constant} .$$

The constant R_{mk} represents the uncertainties in the contact area per collision and contact time.

In situations where either r_{pm} or r_{pk} becomes very small, Eq. (III-132) produces heat-transfer rates larger than those that can occur physically. The capability of the larger droplet to reject or receive heat is limited regardless of the heat-transfer capability of the smaller droplet. These limits are

$$(q_{HLmLk})_{\max} = 15 R_{mk} \alpha_{Lk} \frac{k_{Lk}}{r_{pk}^2} (T_{Lk} - T_{Lm}) ,$$

(III-133)

for small r_{pm} and

$$(q_{HLmLk})_{\max} = 15 R_{mk} \alpha_{Lm} \frac{k_{Lm}}{r_{pm}^2} (T_{Lk} - T_{Lm}) ,$$

(III-134)

for small r_{pk} .

d. Structure-Vapor Heat Transfer. Although heat transfer to the vapor field occurs only during the vaporization/condensation process described in Sec. III.D.6,* the heat-transfer coefficients and interfacial areas for the phase-transition model must be determined. The external surface associated with the fuel pin and the external surface associated with the subassembly can-wall are the structure surfaces available for phase transition. These surface areas result from the region input parameters and the subsequent determination of the structure geometry. For phase transitions, these structure surface areas are modified to represent the expected flow-regime effects more closely. First, if the vapor-volume fraction in the flow channel is less than an input fraction, α_{GSR} , the vapor field is assumed unable to contact the structure surfaces; thus, phase transition on the structure surfaces is inhibited. This capability is included to prevent structure-associated phase transitions when small amounts of vapor are present. Second, to allow a continuous change as the vapor-volume fraction increases, the area available for this mode of phase transition increases linearly with the vapor-volume fraction. Thus, for the pin structure, the nonzero values for A_{Gsk} are

$$A_{Gsk} = \frac{\alpha_G - \alpha_{GSR} (1 - \alpha_S)}{(1 - \alpha_S)(1 - \alpha_{GSR})} A_P \quad \text{if} \quad \frac{\alpha_G}{1 - \alpha_S} > \alpha_{GSR} \quad , \quad (\text{III-135})$$

where A_P is the external surface area associated with the fuel pin. A corresponding decrease occurs for the external surface area associated with the can-wall. Both of these areas are multiplied by an input parameter called SCRC (structure condensation reduction coefficient). It accounts for the apparent exaggeration in structure condensation by the present SIMMER-II model when compared to in-pile LMFBR experiments. A typical value for SCRC is one-tenth.

*A special case of direct structure-to-vapor heat transfer, when the structure is hot and the liquid is limited, is given in Appendix E.

The structure-side heat-transfer coefficient is determined from the region input parameters and the determination of the structure geometry. The vapor-side heat-transfer coefficient is based on a generalized form of standard turbulent flow heat-transfer correlations:

$$h_{GS} = \frac{k_G}{D_h} (C_{GS1} Re_G^{C_{GS2}} Pr_G^{C_{GS3}} + C_{GS4}) \quad , \quad (III-136)$$

where

k_G is the vapor-mixture thermal conductivity,

D_h is the hydraulic diameter for the structure geometry,

C_{GS1} , C_{GS2} , C_{GS3} , C_{GS4} are input correlation coefficients,

$Re_G = \frac{\rho_G |\vec{V}_G| D_h}{\mu_G}$ is the vapor-field Reynolds number, and

$Pr_G = \frac{\bar{c}_{vG} \mu_G}{k_G}$ is the vapor-mixture Prandtl number.

e. Liquid-Vapor Heat Transfer. The surface area of the interface between the liquid component and the vapor field is determined from the liquid droplet radius. However, for small vapor-volume fractions, the assumption of droplets in a continuous vapor field is not valid. Therefore, for vapor-volume fractions less than half the flow volume fraction, the vapor is assumed to exist as bubbles in a continuous liquid field. However, the calculated droplet radii are used as the bubble radii, and the interfacial area is determined to be

$$A_{Lm} = \frac{3\alpha_{Lm}}{r_{pm}} \quad , \quad \text{if } \alpha_G > \frac{(1 - \alpha_S)}{2} \quad , \quad \text{or} \quad (III-137)$$

$$A_{Lm} = \frac{3\alpha_{Lm}}{r_{pm}} \frac{\alpha_G}{\alpha_L} \quad , \quad \text{if } \alpha_G \leq \frac{(1 - \alpha_S)}{2} \quad . \quad (III-138)$$

The vapor-side heat-transfer coefficient is defined to be the value for flow around a single sphere or

$$h_{GLm} = \frac{k_G}{2r_{pm}} (C_{GL1} Re_{GLm}^{C_{GL2}} Pr_G^{C_{GL3}} + C_{GL4} + 2) , \quad (III-139)$$

where

k_G is the average thermal conductivity of the vapor mixture,

C_{GL1} , C_{GL2} , C_{GL3} , C_{GL4} , are input correlation coefficients,

$$Re_{GLm} = \frac{\rho_G |\vec{V}_G - \vec{V}_L| 2r_{pm}}{\mu_G} \quad \text{is the Reynolds number used for vapor flow around a sphere of liquid-field energy component m,}$$

μ_G is the average viscosity of the vapor mixture,

$$Pr_G = \frac{\bar{c}_{vG} \mu_G}{k_G} \quad \text{is the vapor mixture Prandtl number, and}$$

$\bar{c}_{vG}$ is the average specific heat of the vapor mixture.

In the limit of no velocity difference between the liquid and vapor fields, Eq. (III-139) reduces to the result for pure conduction if C_{GL4} is set to zero. The liquid-side heat-transfer coefficient is also determined for pure conduction. In SIMMER-II, it is calculated as

$$h_{LGm} = \frac{k_{Lm}}{0.2 r_{pm}} , \quad (III-140)$$

where k_{Lm} is the thermal conductivity of liquid-field energy component m.

The factor of 0.2 included in the denominator accounts for the median radius of the mass in the droplet, approximately 80% of the droplet radius. In other situations, it may be desirable to have a smaller thermal penetration distance.

5. Momentum-Exchange Models. Momentum exchange between the fields is caused by drag forces and mass exchange. This section describes the momentum-exchange functions that model the drag forces between the structure and liquid fields, between the structure and vapor fields, and between the liquid and vapor fields.

a. Momentum Exchange with the Structure Field. The momentum-exchange functions for drag forces on the structure components use single-phase correlations with a two-phase flow friction multiplier. The friction factor¹⁶ for single-phase flow is defined as if each material were flowing alone:

$$f = \frac{|\Delta p_{SF}|}{4\left(\frac{L}{D}\right)\left(\frac{\rho j^2}{2}\right)}, \quad (\text{III-141})$$

where

f is the friction factor,

$|\Delta p_{SF}|$ is the single-phase pressure drop over a distance given by L ,

D is the hydraulic diameter,

ρ is the single-phase microscopic density, and

j is the fluid superficial velocity (defined below).

The transition from laminar to turbulent flow is assumed to occur at a Reynolds number of 2000. The laminar-flow single-phase friction factor for liquid-structure drag is

$$f_L = \frac{16}{Re_L}, \quad Re_L \leq 2000, \quad (\text{III-142})$$

$$\text{where } Re_L = \frac{D_h |\vec{j}_L| \rho_L}{\sum_m \alpha_m \mu_m}, \quad (\text{III-143})$$

and

D_h is the hydraulic diameter,

$|\vec{j}_L|$ is the magnitude of the superficial liquid-field velocity,

$\bar{\rho}_L$ is the macroscopic liquid-field density,

α_{Lm} is the volume fraction of liquid-field density component m, and

μ_m is the viscosity of liquid-field density component m.

The definition for the liquid superficial velocity is

$$\vec{j}_L = \vec{V}_L \frac{\alpha_L}{1 - \alpha_S} , \quad (III-144)$$

where $\vec{V}_L$ is the real (computed) liquid velocity.

The turbulent-flow single-phase friction factor is given by

$$f_L = f_{LS1} Re_L^{f_{LS2}} , \quad Re_L \geq 2000 \quad (III-145)$$

where f_{LS1} and f_{LS2} are input constants for which typical values are $f_{LS1} = 0.046$ and $f_{LS2} = -0.2$.

For high Reynolds numbers, the friction factor approaches zero when Eq. (III-145) is used. However, the friction factor physically approaches a constant that is dependent on the flow geometry and the surface smoothness. To account for this lower limit, a minimum friction factor f_{Lmin} is input. If the calculated friction factor obtained from Eq. (III-142) or (III-145) is less than f_{Lmin} , then the friction factor f_L is reset to f_{Lmin} .

In this formulation, the particle viscosity has been defined to simulate "blockage" formation for significant concentrations of particles. In the Reynolds number evaluation [Eq. (III-143)] particles are assumed to act in a similar fashion such that

$$\mu_p = \mu_{L6} = \mu_{L7} = \mu_{L8} = \mu_p , \quad \text{and} \quad (III-146)$$

$$\alpha_p = \sum_{m=6}^8 \alpha_{Lm} , \quad (III-147)$$

where

μ_p is an input particle viscosity and

α_p is a particle volume fraction.

If structure exists to define a hydraulic diameter, a high concentration of high viscosity particles will reduce liquid flow rates greatly. Additional mechanisms whereby particles can increase liquid friction are described in Sec. III.D.5.e.

The friction factor for the vapor-structure drag is calculated in a manner similar to the calculation of the liquid-structure friction factor. The laminar flow single-phase friction is

$$f_G = \frac{16}{Re_G} , \quad Re_G \leq 2000 , \quad (III-148)$$

$$\text{where } Re_G = \frac{D_h |\vec{j}_G| \bar{\rho}_G}{\alpha_G \mu_{mix}} , \quad \text{and} \quad (III-149)$$

$|\vec{j}_G|$ is the magnitude of the superficial vapor velocity,

$\bar{\rho}_G$ is the vapor-field macroscopic density,

α_G is the vapor-field volume fraction, and

μ_{mix} is the vapor mixture average viscosity.

The definition for the vapor superficial velocity is

$$\vec{j}_G = \vec{v}_G \frac{\alpha_G}{1 - \alpha_S} , \quad (III-150)$$

where $\vec{v}_G$ is the real (computed) vapor velocity.

The single-phase turbulent-flow friction factor for vapor-structure drag is

$$f_G = f_{GS1} Re_G^{f_{GS2}} , \quad Re_G > 2000 , \quad (III-151)$$

where f_{GS1} and f_{GS2} are input constants.

The minimum vapor friction factor is the input constant f_{Gmin} .

The two-phase pressure drop, $|\nabla p|_{Tp}$, is evaluated from the Martinelli correlation, where two-phase friction-factor multipliers are defined by

$$\phi_G^2 = \frac{|\nabla p|_{Tp}}{|\nabla p_G|} \quad , \quad \text{and} \quad (\text{III-152})$$

$$\phi_L^2 = \frac{|\nabla p|_{Tp}}{|\nabla p_L|} \quad , \quad (\text{III-153})$$

$$\text{where } \nabla p_G = - \frac{2f_G}{D_h} \rho_G |\vec{j}_G| \vec{j}_G \quad , \quad \text{and} \quad (\text{III-154})$$

$$\nabla p_L = - \frac{2f_L}{D_h} \rho_L |\vec{j}_L| \vec{j}_L \quad . \quad (\text{III-155})$$

By taking the ratio of the two-phase friction-factor multipliers, a dimensionless parameter X^2 can be defined as

$$X^2 = \frac{\phi_G^2}{\phi_L^2} = \frac{f_L \rho_L j_L^2}{f_G \rho_G j_G^2} \quad . \quad (\text{III-156})$$

From dimensional analysis and using Eq. (III-156), we can show that

$$\phi_L^2 = [1 + (\frac{1}{X^2})^{1/n}]^n \quad , \quad \text{and} \quad (\text{III-157})$$

$$\phi_G^2 = [1 + (X^2)^{1/n}]^n \quad , \quad (\text{III-158})$$

where n must be determined from a simplified analytical model or from correlation to experimental data. Wallis¹⁷ suggests that a reasonable correlation for all flow regimes is given by $n = 3.5$; but $n = 4$ gives a better correlation in the turbulent-turbulent regime. In SIMMER-II, we assume

$$\begin{aligned} n &= 3.5 \text{ for either } Re_G \text{ or } Re_L \text{ less than } 2000 \text{ and} \\ n &= 4 \text{ for both } Re_G \text{ and } Re_L \text{ greater than } 2000. \end{aligned}$$

However, the Martinelli correlation¹⁸, does not give any information as to how much of the pressure drop should be attributed to the separate liquid and vapor momentum fields. If we add the liquid and vapor field momentum equations and assume a system without phase change, acceleration, or body force effects (such as gravity), we obtain

$$(\alpha_L + \alpha_G) \nabla p_{Tp} = -(K_{LS} \vec{v}_L + K_{GS} \vec{v}_G) \quad . \quad (\text{III-159})$$

This suggests that particularly attractive choices for K_{LS} and K_{GS} are

$$K_{LS} = \frac{|\nabla p_L| \phi_L^2}{|\vec{v}_L|} [\alpha_L \min(2R, 1) + \alpha_G \max(0, 2R-1)] \quad , \quad \text{and} \quad (\text{III-160})$$

$$K_{GS} = \frac{|\nabla p_G| \phi_G^2}{|\vec{v}_G|} [\alpha_L \max(0, 1-2R) + \alpha_G \min(2-2R, 1)] \quad , \quad (\text{III-161})$$

where R is an input parameter such that $0 \leq R \leq 1$.

In this case, $R = 0$ attributes the total pressure drop to the vapor field, $R = 1$ attributes the total pressure drop to the liquid field, and $R = 0.5$ yields

$$\alpha_L \nabla p_{Tp} = -K_{LS} \vec{v}_L \quad , \quad \text{and} \quad (\text{III-162})$$

$$\alpha_G \nabla p_{Tp} = -K_{GS} \vec{v}_G \quad . \quad (\text{III-163})$$

For SIMMER variables, the expressions for K_{LS} and K_{GS} are

$$K_{LS} = \frac{2f_L}{D_h} \bar{\rho}_L |\vec{v}_L| \Phi_L^2 \frac{\alpha_L}{(1 - \alpha_S)^2} [\alpha_L \min(2R, 1) + \alpha_G \max(0, 2R-1)] \quad , \quad \text{and} \quad (\text{III-164})$$

$$K_{GS} = \frac{2f_G}{D_h} \bar{\rho}_G |\vec{v}_G| \Phi_G^2 \frac{\alpha_G}{(1 - \alpha_S)^2} [\alpha_L \max(0, 1 - 2R) + \alpha_G \min(2 - 2R, 1)] \quad . \quad (\text{III-165})$$

For single-phase flow, the vapor-structure momentum exchange function is set to zero, and all the momentum drag on structure results from the liquid field.

b. Liquid-Field--Vapor-Field Momentum Exchange. SIMMER-II uses a modified liquid-vapor drag correlation for dispersed flow in which the momentum exchange function has the form

$$K_{GLm} = \frac{3\bar{\rho}_G \alpha_{Lm}}{2\alpha_G \left[\frac{\alpha_G}{1 - \alpha_S} \right]^{\lambda-1} \left[\frac{\alpha_G}{1 - \alpha_S} - \alpha_o \right] r_{pm}^2} \left(3\nu_G + \frac{r_{pm} C_D}{4} |\vec{v}_{LG}| \right) \quad , \quad (\text{III-166})$$

where

$\bar{\rho}_G$ is the vapor-field macroscopic density,

α_{Lm} is the liquid volume fraction for liquid-field energy component m ,

α_G is the vapor volume fraction,

α_0 is an input parameter such that $\alpha_0(1-\alpha_S)$ is the fraction of the flow volume fraction below which single-phase calculations are performed,

α_S is the structure volume fraction,

λ is an arbitrary input parameter (PIEE),

ν_G is the vapor kinematic viscosity,

C_D is a form drag coefficient, input as CDR, and

$\vec{v}_{LG} = \vec{v}_L - \vec{v}_G$ is the relative velocity between the liquid field and the vapor field.

This correlation results from a combination of Stokes flow over a sphere for low Reynolds numbers and a form drag correlation for spheres for high Reynolds numbers. The $(\alpha_G/(1 - \alpha_S) - \alpha_0)$ term is included in the denominator for continuity with single-phase calculations, and the $[\alpha_G/(1 - \alpha_S)]^{\lambda-1}$ term can be used to simulate drag in highly packed particle beds. The kinematic viscosity for Eq. (III-166) is determined from μ_G/ρ_G , with μ_G obtained from Eq. (III-85).

In KACHINA and SIMMER-I, only one particle radius was calculated; therefore K_{GL} was evaluated for only one particle size. In SIMMER-II, the total force on the liquid field is assumed to be equal to the sum of the forces on the individual liquid-field components, or

$$K_{GL} = \sum_{m=1}^{NCLE} K_{GLm} \quad , \quad \text{or} \quad (III-167)$$

$$K_{GL} = \frac{3 \bar{\rho}_{333G}}{2 \alpha_G \left(\frac{\alpha_G}{1 - \alpha_S} \right)^{\lambda-1} \left[\frac{\alpha_G}{1 - \alpha_S} - \alpha_0 \right]} \left[3 \nu_G \sum_{m=1}^{NCLE} \frac{\alpha_{Lm}}{r_{pm}^2} + \frac{C_D}{4} |\vec{v}_{LG}| \sum_{m=1}^{NCLE} \frac{\alpha_{Lm}}{r_{pm}} \right] \quad . \quad (III-168)$$

For single-phase flow calculations in mesh cells, the liquid and vapor fields are assumed to move together. The liquid-vapor momentum-exchange function is set to a large number to force the relative motion to zero. For the SI

units used in SIMMER-II, a value for K_{GL} of 10^{10} is sufficient to force this condition.

c. Drag Heating. The viscous shear that causes drag between the fluids and the structure also causes mechanical energy to be converted into thermal energy. In SIMMER-I, this drag heating from liquid-vapor drag and structure-vapor drag was assigned to the vapor field; the drag heating from structure-liquid drag was split arbitrarily among the liquid-field components in proportion to the local liquid-field component mass fractions. In SIMMER-II, structure-liquid drag heating is ignored. The total drag heating for the remaining drag components is calculated in the same manner as in SIMMER-I; that is,

$$q_{KGS} = K_{GS} |\vec{v}_G|^2, \text{ and} \quad (\text{III-169})$$

$$q_{KGL} = K_{GL} |\vec{v}_{LG}|^2, \quad (\text{III-170})$$

where

q_{KGS} is the total drag heating because of drag between the structure field and the vapor field and

q_{KGL} is the total drag heating because of drag between the liquid field and the vapor field.

The drag heating for each drag component is divided among the energy components in proportion to the heat-transfer coefficients at the interface where the drag heating takes place. Thus, the drag between the liquid and the vapor fields causes an energy transfer to the vapor field of

$$q_{KG(KGL)} = \left[\frac{\sum_{m=1}^{NCLE} h_{GLm} A_{Lm}}{\sum_{m=1}^{NCLE} (h_{GLm} + h_{LGm}) A_{Lm}} \right] q_{KGL}, \quad (\text{III-171})$$

and an energy transfer to each liquid-field energy component of

$$q_{KLm}(KGL) = \left[\frac{h_{LGm} A_{Lm}}{\sum_{m=1}^{NCLE} (h_{GLm} + h_{LGm}) A_{Lm}} \right] q_{KGL} \quad (III-172)$$

Similarly, vapor-structure drag causes drag heating of the vapor mixture, which is calculated as

$$q_{KG}(KGS) = \left[\frac{\sum_{k=1}^2 A_{Sk} h_{GS}}{\sum_{k=1} A_{Sk} (h_{GS} + h_{Sk})} \right] q_{KGS} \quad (III-173)$$

Finally, the drag heating of the structure-field energy components from vapor-structure drag is modeled as

$$q_{KSk}(KGS) = \left[\frac{A_{Sk} h_{Gk}}{\sum_{k=1} A_{Sk} (h_{GS} + h_{Sk})} \right] q_{KGS} \quad (III-174)$$

The only structure-field components participating in this division of energy are the two components that the structure-configuration model determined to be exposed to the flow fields.

d. Orifice Pressure Drop. The subassemblies for LMFBRs of the current US design have a high orifice pressure drop. This pressure drop is modeled in SIMMER-II by increasing the drag between the structure field and the vapor and liquid fields at the bottom boundary of the calculative mesh. The model is intended to be consistent with the usual textbook definition of an orifice coefficient. For the liquid field, the exchange function is increased by

$$K_{LORF} = \frac{C_{ORF} \bar{p}_L |\vec{v}_L|}{\Delta z_1} , \quad (III-175)$$

where C_{ORF} is the radially dependent orifice coefficient. A similar expression is used for the vapor field:

$$K_{GORF} = \frac{C_{ORF} \bar{p}_G |\vec{v}_G|}{\Delta z_1} . \quad (III-176)$$

For added flexibility, an input parameter, O_L , is used to change the location of the orifice coefficient. If O_L is input as 0, Eqs. (III-175) and (III-176) are evaluated and the friction is increased on the lower boundary of the first node. If $O_L = j$, with $j \geq 1$, the exchange functions are increased by

$$K_{LORF} = \frac{C_{ORF} (\bar{p}_{L,j} \Delta z_j + \bar{p}_{L,j+1} \Delta z_{j+1}) |\vec{v}_{L,j+1/2}| \alpha_L}{(\alpha_{L,j} \Delta z_j + \alpha_{L,j+1} \Delta z_{j+1}) \Delta z_j} , \quad (III-177)$$

and

$$K_{GORF} = \frac{C_{ORF} (\bar{p}_{G,j} \Delta z_j + \bar{p}_{G,j+1} \Delta z_{j+1}) |\vec{v}_{G,j+1/2}| \alpha_G}{(\alpha_{G,j} \Delta z_j + \alpha_{G,j+1} \Delta z_{j+1}) \Delta z_j} . \quad (III-178)$$

where

$$\alpha_G = \min (\alpha_{G,j} , \alpha_{G,j+1}) ,$$

$$\alpha_S = \max (\alpha_{S,j} , \alpha_{S,j+1}) ,$$

$$\alpha_L = 1 - \alpha_G - \alpha_S ,$$

and the friction is increased on the top boundary of a cell. In other words, the effective momentum exchange coefficients in evaluating the velocity change at the $j + 1/2$ interface are

$$K_{LS,j+1/2} = 2[K_{LORF} \Delta z_j + 1/2(K_{LS,j} \Delta z_j + K_{LS,j+1} \Delta z_{j+1})]/(\Delta z_j + \Delta z_{j+1}) \quad , \quad (III-179)$$

$$K_{GS,j+1/2} = 2[K_{GORF} \Delta z_j + 1/2(K_{GS,j} \Delta z_j + K_{GS,j+1} \Delta z_{j+1})]/(\Delta z_j + \Delta z_{j+1}) \quad . \quad (III-180)$$

Actually Eqs. (III-179) and (III-180) also are used when O_L is 0, but in this case $\Delta z_j = \Delta z_{j+1} = \Delta z_1$, $K_{LS,j} = K_{LS,j+1} = K_{LS,1}$, $K_{GS,j} = K_{GS,j+1} = K_{GS,1}$, and the boundary exchange coefficients are $K_{LS,1/2}$ and $K_{GS,1/2}$, thus Eqs. (III-177) and (III-178) reduce to (III-175) and (III-176) respectively. In view of the association of K_{LORF} and K_{GORF} with a cell interface rather than a particular cell, these variables have not been included in the standard SIMMER full print edit.

e. Particle Jamming Restrictions. The method for fitting particles into the liquid Reynolds number was described in Sec. III.D.5.a. However, particles also should be limited in moving into a cell before all the void volume disappears because of the inability of solids to deform. This section will delineate these restrictions.

The algorithm multiplies the liquid frictional exchange coefficient at a momentum-cell boundary by a calculated multiplier. An illustration of such a coefficient is given by Eq. (III-179). An initial trial value of the multiplier is given by the maximum of two expressions. The first expression for the particle multiplier, P_M , is a simple step function when the average particle diameter of an acceptor cell exceeds the channel hydraulic diameter and sufficient material actually exists to produce a particle of the indicated size in a cell. In the axial direction this expression is

$$P_M = 10^{10} \quad \text{if} \quad \frac{2r_f}{D_h} > 1 \quad \text{and if}$$

$$2r_f (1 - a_S) \rho_{A6} < (\Delta z)_j [\bar{p}_{A6} + \bar{p}_{A7}] \quad ; \quad \text{also} \quad (III-181)$$

$$P_M = 10^{10} \quad \text{if} \quad \frac{2r_s}{D_h} > 1 \quad \text{and if}$$

$$2 r_s (1 - \alpha_S) \rho_{L8} < (\Delta z)_j \bar{p}_{L8} ; \quad (\text{III-182})$$

otherwise, $P_M = 1$. In this expression, r_f is the solid fuel particle radius (input for a mesh region or on a node-by-node basis), r_s is the solid steel particle radius (input like r_f), D_h is the cell hydraulic diameter, and $(\Delta z)_j$ is the axial mesh spacing of the acceptor cell. A similar expression exists for the radial direction.

The second expression for the exchange function multiplier is based on a packing fraction limitation. This expression requires that P_M be continuous and become large as the maximum packing fraction is approached in an acceptor cell. The chosen expression is

$$P_M = \frac{1}{\max \left(1 - C_2 \frac{\alpha_p}{1 - \alpha_S}, 10^{10} \right)} , \quad (\text{III-183})$$

where C_2 is the inverse of the maximum packing fraction (input as PKFMAX), α_p is the particle fraction from Eq. (III-147), and α_S is the structure volume fraction.

This trial value can be reduced to insure pressure relief with expanding liquids (such as melting cladding). The approach is to divide the local gravity head by the pressure gradient, thus representing the inverse of the unbalanced forces, U , tending to compact or disrupt the blockage. That is,

$$U = \min \left[\frac{\bar{p}_L g}{|\nabla p|} , 1 \right] . \quad (\text{III-184})$$

Because higher pressure gradients are much more likely to disrupt (move) particle blockages, the particle multiplier is taken as depending exponentially on U , or

$$P_M = \min [P'_M, 10^{10} U^5] \quad U > .01 \quad ,$$

and

(III-185)

$$P_M = 1 \quad U \leq .01 \quad ,$$

where P'_M is the maximum trial value from Eqs. (III-181) to (III-183).

f. Plug Modeling. The expansion mechanics of high-temperature core material after a severe core disruptive accident in an LMFBR may produce loads on the rotating plugs in the reactor head and the discharge of materials to the Reactor Containment Building. The SIMMER-II code can address the coupled problem of plug acceleration and discharge if plug mechanics is inserted to accelerate the plug out of a nonflow region representing the reactor head. A special version of SIMMER can be compiled using a *IF DEF, PLUG option containing plug mechanics and special editing.

The plug motion is modeled very simply by Newton's second law. The acceleration is computed from the mass of the plug and the integrated forces acting on it. SIMMER-II computes the pressure at the lower surface of the plug, whereas an opposing pressure of 0.1 MPa is assumed to act on the upper surface of the plug. Of course, gravity is included. A breakaway force, which represents the restraint from bolts or shear keys, is used as a threshold for initiation of plug motion. The velocity and displacement of the plug are obtained through successive time integrations of the acceleration and the velocity. This motion is computed simultaneously with the other momentum exchange functions and influences the fluid dynamics methods as described in Sec. IV.

6. Mass-Exchange Models. Several different modes of mass exchange are modeled in SIMMER-II. The structure-field energy components can melt (or break up) and transfer mass to the liquid field. Similarly, solid particles in the liquid field can melt and transfer mass to the liquid components of the liquid field. However, the liquid components can freeze with mass transferred to the solid components in both the structure and liquid fields. Vaporization and condensation can occur between the liquid components and the vapor-field components. Finally, fission gas mass transfer can occur from the intragranular to the intergranular fission gas component and then to the vapor field.

a. Solid-Component Breakup Rates. The current structural breakup model depends on the energy transferred to a structure surface, the energy of the structural component, and the structural components present. First, the non-

equilibrium melting/freezing model can lead to melting of a structure surface. The melting energy flux, q_{ISk} , is obtained by summing Eq. (III-127) and Eq. (III-128) where the interface temperature has been set to $T_{Melt,Fuel}$ for a fuel surface and to $T_{Melt,Steel}$ for a steel surface. This gives

$$q_{ISk} = \sum_{m=1}^4 q_{ILmSk} + q_{ISkL} \quad , \quad (III-186)$$

where q_{ISk} is negative. (There is net energy flow from the surface to the liquid field.) The melting rate is estimated from

$$\Gamma_{SkLm} = -q_{ISk} / [e_{Liq,M} - e_{Sk}] \quad , \quad (III-187)$$

where $e_{Liq,M}$ is the liquidus energy for material M, which must correspond to structural energy component k and liquid energy component m. The molten material appears as liquid fuel or steel. The energy associated with this mass transfer appearing in the liquid field is

$$q_{\Gamma LmM} = \Gamma_{SkLm} e_{Liq,M} \quad , \quad (III-188)$$

and, for conservation of energy, the energy taken out of the structure field is

$$q_{\Gamma SkM} = -\Gamma_{SkLm} e_{Sk} \quad . \quad (III-189)$$

In addition, the user can specify, if desired, for additional structure to be transferred to the liquid momentum field as particles. The fraction of the material that is transferred as liquid is specified by the user as x_{Fk} . Consequently, the total rate of mass transfer is Γ_{SkLm}/x_{Fk} , and the particle transfer is $\Gamma_{SkLm}(1-x_{Fk})/x_{Fk}$. These particles are formed at the structure energy, e_{Sk} . If there is not enough structure, the particle mass flux is reduced correspondingly.

Second, the current SIMMER-II code will calculate breakup mass-transfer rates from the advanced-time structure-field energy-component specific internal energies, $\tilde{e}_{Sk}^{n+1}$. These are estimated from the heat-transfer calculation described in Sec. IV and updated with vaporization-condensation and drag heating contributions. This model includes control material as well as fuel and steel. The structure density still available for breakup is

$$\tilde{\rho}_{Sk}^n = \tilde{\rho}_{Sk}^n - (\Gamma_{SkLm}/x_{Fk} - \Gamma_{LmSk}) \Delta t \quad , \quad (\text{III-190})$$

where k potentially covers the first four structure components and Γ_{LmSk} comes from the nonequilibrium freezing contribution [Eq. (III-193)]. To determine the breakup rate, the $\tilde{e}_{Sk}^{n+1}$ are compared with the breakup energy given by

$$e_{Fk} = e_{Sol,M} + f_{Fk} h_{f,M} \quad , \quad (\text{III-191})$$

where f_{Fk} is the input fraction of the heat of fusion through which the component energy is increased before breakup of the component. When $\tilde{e}_{Sk}^{n+1}$ exceeds e_{Fk} , breakup of component k is assumed to occur, and additional mass transfer to the liquid field takes place. Mass transfer still occurs as a mixture of liquid and particles, but x_{Fk} should be greater than f_{Fk} . The liquid mass transfer rate per unit volume is added to that calculated in Eq. (III-187) to obtain

$$\Gamma_{SkLm} = \frac{\tilde{\rho}_{Sk}^n}{\Delta t} (\tilde{e}_{Sk}^{n+1} - e_{Sol,M})/h_{f,M} + \Gamma'_{SkLm} \quad , \quad (\text{III-192})$$

where Γ'_{SkLm} is the mass transfer rate calculated by Eq. (III-187) if nonzero. The particulate transfer is calculated as in the nonequilibrium model, and the liquid still appears in the liquid field at the liquidus energy. However, the liquid removed from the structure field is taken out at the liquidus energy, leaving the remaining structure at the solidus energy. Particles are transferred at the solidus energy. If $(\tilde{e}_{Sk}^{n+1} - e_{Sol,M})/h_{f,M} > x_{Fk}$, all the remaining structure of component k is transferred with a reduction in particulate to con-

serve energy. If $\bar{e}_{Sk}^{n+1} > e_{Liq,M}$, the newly formed liquid is initialized with energy $\bar{e}_{Sk}^{n+1}$.

Third, complete breakup of a structural component can occur under some circumstances. Breakup of fabricated fuel into particles occurs if both cladding and can-wall disappear, or if cladding disappears and no fabricated fuel exists in the cell below the cell being examined. Fuel crust breakup occurs when a can-wall first disappears. Crust regrowth can occur because of the can-wall set-over algorithm; however, a second crust breakup occurs when no neighboring can-walls exist. In all these cases, the particles are initialized with the original structure energy.

b. Liquid-Component Freezing Rates. The freezing model of SIMMER-II is the inverse of the breakup model except that the freezing model is less general. The nonequilibrium model calculates q_{ISk} in the same fashion; however, q_{ISk} is positive in freezing. The freezing rate is estimated from

$$\Gamma_{LmSk} = q_{ISk} / e_{Lm} - e_{Sol,M} \quad (III-193)$$

The pin structure cannot accept a crust. Consequently, any fuel mass flux to the pin surface is changed to a particle mass flux, which is identified as Γ_{LlL5} , a mass rate per unit volume of liquid fuel to solid fuel particles. Also, with the nonequilibrium model, no steel is transferred to/from a can-wall set-over from another cell. (The logic has not been coded.) If such steel transfer is suggested, the energy flows are recalculated using the interface temperature implied from Eq. (III-130); that is, an equilibrium model is used. With the nonequilibrium model, the solid material can be either fuel or steel. The energy associated with this mass transfer that appears in the structure field is

$$q_{\Gamma SkF} = \Gamma_{LmSk} e_{sol,M} \quad (III-194)$$

The energy taken out of the liquid field is

$$q_{\Gamma LmF} = -\Gamma_{LmSk} e_{Lm} \quad (III-195)$$

After the nonequilibrium transfers are calculated, the advanced time liquid-component energy, $\tilde{e}_{Lm}^{n+1}$, is compared with the liquidus energy. If $\tilde{e}_{Lm}^{n+1}$ is less, then sufficient solid material is taken from the liquid component to leave the liquid energy equal to the liquidus energy. Hence, the extra mass transferred per unit volume for freezing of a liquid component is given by

$$\Delta t \Gamma_{LmF} = \tilde{\rho}_{Lm}^n \left(\frac{e_{Liq,M} - \tilde{e}_{Lm}^{n+1}}{h_{f,M}} \right), \quad (III-196)$$

where

$$\tilde{\rho}_{Lm}^n = \bar{\rho}_{Lm}^n - \sum_k (\Gamma_{LmSk} - \Gamma_{SkLm}) \Delta t \Gamma_{LmGm} \Delta t,$$

Γ_{LmSk} and Γ_{SkLm} come from Eq. (III-193) and (III-187), and

Γ_{LmGm} is a vaporization rate for liquid component m. (Condensation is not included because of the data management in the current version of SIMMER-II.)

Here Γ_{LmF} is a mass-transfer rate per unit volume because of freezing of liquid component m, and $\bar{\rho}_{Lm}^n$ is the current macroscopic density of liquid component m. Again, if the mass transfer from a liquid component in a single time step exceeds the total liquid mass present for that component, then the mass-transfer rate is reset to eliminate only the mass present by

$$\Delta t \Gamma_{LmF} = \tilde{\rho}_{Lm}^n. \quad (III-197)$$

The energy loss from the liquid component because of mass transfer is summed to be

$$q_{\Gamma LmF} = -\Gamma_{LmF} e_{Sol,M} + q'_{\Gamma LmF}, \quad (III-198)$$

where $q'_{\Gamma LmF}$ was calculated by Eq. (III-195).

As liquid fuel and steel freeze, the solid material can be transferred into several components. For example, liquid fuel can freeze and become frozen fuel attached to can-wall structure, or it can freeze into solid particles carried

with the liquid field. Liquid steel can freeze to the cladding or the can-wall, or it can freeze into solid particles in the liquid field. Equation (III-196) does not indicate to which solid components these liquid components transfer mass. This partition of the freezing rates is done in proportion to the heat-transfer rates of the various components.

As the heat transfer to the liquid-fuel component is computed, a cumulative total is kept for the heat transfer to the can-wall structure and for heat transfer to the fuel-pin structure and the other liquid-field components. The freezing rate of liquid fuel to structure-field frozen fuel then is calculated as

$$\Gamma_{L1S2} = \frac{q_{HSCL1} \Gamma_{L1F}}{q_{HSPL1} + q_{HSCL1} + q_{HLL1}} + \Gamma'_{L1S2} \quad , \quad (III-199)$$

where

Γ_{L1S2} is the mass-transfer rate from liquid to frozen fuel,

Γ'_{L1S2} is the rate calculated in Eq. (III-193),

q_{HSPL1} is the heat-transfer rate from liquid fuel to the pin structure,

q_{HSCL1} is the heat-transfer rate from liquid fuel to the can wall structure, and

q_{HLL1} is the total heat-transfer rate from the liquid fuel to the other liquid-field components.

The structure-configuration model will set the necessary component for the pin and can-wall surface. Similarly, the freezing rate of liquid fuel into solid particles is

$$\Gamma_{L1L5} = \frac{(q_{HSPL1} + q_{HLL1}) \Gamma_{L1F}}{q_{HSPL1} + q_{HSCL1} + q_{HLL1}} + \Gamma'_{L1L5} \quad . \quad (III-200)$$

For the freezing of liquid steel into solid steel, three heat-transfer paths determine the mass-transfer paths. The three ultimate destinations of the solid steel are the cladding, the can-wall, and the solid steel particles. Individual cumulative totals are kept for the heat transfer from liquid steel to

the cladding, to the can-wall, and to the other liquid components. Then the mass-transfer rates are defined as

$$\Gamma_{L2S3} = \frac{q_{HS3L2} \Gamma_{L2F}}{q_{HS3L2} + q_{HS4L2} + q_{HLL2}} + \Gamma'_{L2S3} , \quad (\text{III-201})$$

$$\Gamma_{L2S4} = \frac{q_{HS4L2} \Gamma_{L2F}}{q_{HS3L2} + q_{HS4L2} + q_{HLL2}} + \Gamma'_{L2S4} , \quad \text{and} \quad (\text{III-202})$$

$$\Gamma_{L2L6} = \frac{q_{HLL2} \Gamma_{L2F}}{q_{HS3L2} + q_{HS4L2} + q_{HLL2}} , \quad (\text{III-203})$$

where

Γ_{L2F} is the total rate of freezing of liquid steel because of cooling below the liquidus energy,

Γ_{L2S3} is the mass-transfer rate per unit volume of liquid steel to solid cladding,

Γ_{L2S4} is the mass-transfer rate per unit volume of liquid steel to solid can-wall,

Γ'_{L2S3} and Γ'_{L2S4} are the mass-transfer rates calculated with nonequilibrium model, Eq. (III-193),

Γ_{L2L6} is the mass-transfer rate per unit volume of liquid steel to solid steel particles,

q_{HS3L2} is the heat-transfer rate from liquid steel to the pin structure,

q_{HS4L2} is the heat-transfer rate from liquid steel to the can-wall structure, and

q_{HLL2} is the total heat-transfer rate from liquid steel to the other liquid field components.

There is no contribution from the nonequilibrium model for steel particles.

The energy created by mass transfer occurs as a source at the solidus energy to those components to which the mass is transferred. In the case of fuel and steel, this energy transfer is partitioned according to the mass transfer as given by Eqs. (III-199) through (III-203).

Finally, there is no solid particle component for control material in the liquid momentum-field. Frozen control material must be associated with the structure field. If no structure exists in a cell, freezing control material would create a small amount of stationary but isolated solid material. To avoid this problem, liquid control material is not allowed to freeze unless a structure-field component already exists in a cell.

c. Simple Vaporization-Condensation Model. The simple vaporization-condensation model used in SIMMER-II is one where the rate of phase transition is limited by the rate of heat conduction to or from the interface where the phase transition is assumed to occur. The model is similar to the one used in SIMMER-I but has been modified to accept the complications introduced by the additional components and the structure-configuration model of SIMMER-II. In addition, the resulting model equations are solved in a much different manner than in SIMMER-I.

Vaporization and condensation are assumed to occur in two possible modes. In the first mode, the phase transition occurs at the interface between the vapor field and a given liquid-field energy component. The component mass transferred in this mode is assumed to be only the material composing the liquid-field energy component. Thus, fuel vapor is not allowed to condense on liquid sodium. However, if liquid fuel is present, the fuel vapor can condense on the liquid fuel provided the thermodynamic conditions are favorable for such a process.

In the second mode of vaporization and condensation, the phase transition is assumed to occur at the interface between vapor field and a structure surface. For condensation, this phenomenon can be considered as drop-wise condensation. Justification for the model breaks down for vaporization because the mass vaporized at the interface cannot come from the structure surface. However, in SIMMER-II liquid films are assumed to provide the mass vaporized even though the films are not included as part of the structure-vaporization model. Thus, any component species can be involved in the phase transition occurring at a structure interface provided that species exist either in the vapor or liquid fields.

To obtain the phase-transition rate at the interface between the vapor field and a given liquid-field energy component, the heat flows are determined from the interface to the vapor field and to the liquid-field component. The interfacial temperature used to determine these heat flows is assumed to be the saturation temperature at the partial pressure for the vapor species involved in the phase-transition process. Thus, the heat flows are

$$q_{IGmLm} = h_{GLm} A_{Lm} (T_{Sat,M} - T_G) \quad , \text{ and} \quad (III-204)$$

$$q_{ILmGm} = h_{LGm} A_{Lm} (T_{Sat,M} - T_{Lm}) \quad , \quad (III-205)$$

where

q_{IGmLm} is the heat transferred per unit volume from the interface of liquid-field energy component m to the vapor field by phase transition,

q_{ILmGm} is the heat transferred per unit volume from the interface of liquid-field energy component m into the liquid component by phase transition,

h_{GLm} is the vapor-side heat-transfer coefficient for liquid-field energy component m,

h_{LGm} is the liquid-side heat-transfer coefficient for liquid-field energy component m,

A_{Lm} is the surface area per unit volume of liquid-field energy component m,

$T_{Sat,M}$ is the saturation temperature for the partial pressure because of the vapor species of the same composition as liquid-field energy component m,

T_G is the vapor-field temperature, and

T_{Lm} is the temperature of liquid-field energy component m.

The net heat flow from the interface determines the phase-transition rate. If the net heat flow from the interface is negative (that is, $q_{IGmLm} + q_{ILmGm}$ is negative), the excess energy at the interface vaporizes liquid and the positive phase transition rate is

$$\Gamma_{ILmGm} = - \frac{(q_{IGmLm} + q_{ILmGm})}{(h_{\ell g, M})} , \quad (III-206)$$

where

Γ_{ILmGm} is the rate of mass transfer per unit volume from liquid-field energy component m to vapor-field material component m because of phase transition occurring at the liquid-component interface and

$h_{\ell g, M}$ is the heat of vaporization for the component material considered.

If the net heat flow is away from the interface, the excess energy originating at the interface results from condensation, and the positive phase-transition rate is

$$\Gamma_{IGmLm} = \frac{(q_{IGmLm} + q_{ILmGm})}{h_{\ell g, M}} , \quad (III-207)$$

where Γ_{IGmLm} is the mass-transfer rate per unit volume from vapor-field material component m to liquid-field energy component m because of phase transition occurring at the liquid-component interface. structure components are determined in a similar manner. First, the heat flows from the interface between the vapor field and structure components are calculated. These are

$$q_{IGmSk} = h_{GS} A_{GSk} (T_{Sat, M} - T_G) , \quad \text{and} \quad (III-208)$$

$$q_{ISkGm} = h_{Sk} A_{GSk} (T_{Sat, M} - T_{Sk}) , \quad (III-209)$$

where

q_{IGmSk} is the heat transferred per unit volume from the interface of structure-field energy component k to the vapor field from phase transition,

q_{ISkGm} is the heat transferred per unit volume from the interface of structure-field energy component k into the structure component from phase transition,

h_{GS} is the vapor-side heat-transfer coefficient for vapor-to-structure heat transfer,

h_{Sk} is the structure-side heat-transfer coefficient for structure-field energy component k ,

A_{GSk} is the available surface area per unit volume of structure-field energy component k as given by Eq. (III-135) and the associated equation for the can-wall surface, and

T_{Sk} is the temperature of structure-field energy component k .

For a given liquid-field energy component or vapor-field material, the phase-transition rate again is calculated from the net heat flow from the interface. For condensation, these rates are given by

$$(\Gamma_{GmLm})_{ISk} = \frac{(q_{IGmSk} + q_{ISkGm})}{h_{lg,M}}, \quad (III-210)$$

where $(\Gamma_{GmLm})_{ISk}$ is the rate of mass transfer per unit volume of vapor-field material component m to liquid-field energy component m because of phase transitions occurring at the interface of structure-field energy component k . Again, a positive sum implies condensation, whereas a negative value indicates vaporization. If Eq. (III-210) indicates vaporization, the appropriate phase-transition rate is $(\Gamma_{LmGm})_{ISk}$.

The mass transferred during vaporization and condensation also transports energy between the liquid and vapor fields. This transport of energy depends on the direction of the phase transition. For both condensation and vaporization, the phase-transition process is assumed to occur at the local saturation conditions. Thus, the vapor material is interacting in the process at the saturated vapor energy, $e_{vap,M}$, and any vapor entering or leaving the vapor field during the process does so at the saturated vapor energy. Similarly, any liquid material entering or leaving the liquid field during phase transition does so at the condensate energy, $e_{con,M}$. However, there is an additional energy transfer occurring during the phase transition because of the mechanical work done resulting from the liquid-vapor volume change. Thus, for example, the energy transfer to the liquid-field energy component m during condensation is

$$q_{\Gamma LmGm} = [\Gamma_{IGmLm} + \sum_k (\Gamma_{GmLm})_{ISk}] e_{con,M}. \quad (III-211)$$

The energy loss from the vapor is

$$q_{\Gamma GmLm} = -[\Gamma_{IGmLm} + \sum_k (\Gamma_{GmLm})_{ISk}] [e_{vap,M} + (p\Delta v)_M] \quad (III-212)$$

The situation is similar for vaporization except that the signs and the component subscripts on the right side of Eqs. (III-211) and (III-212) are reversed.

d. Multicomponent Phase-Transition Model. An optional vaporization-condensation model in SIMMER-II includes the interaction between the various material components during the phase-transition process. Not only does this model predict the suppression of condensation by noncondensables such as fission gas, but it also predicts the phase-transition rate for a vapor component condensing on the surface of a different material. In SIMMER-II, this model can be used only with the analytic equation of state. In addition, the user may need to adapt the algorithm to the needs of the particular problem to be examined.

The equations for this model are obtained by considering the steady-state boundary layer for the horizontal interface shown in Figs. 4-5. The coordinate system is assumed to move with the interface such that y represents the distance from the interface. Conditions just below the interface are subscripted by "a," whereas those just above it are subscripted by "b". The bulk conditions for the vapor (assumed to be above the boundary layer) are subscripted by " ∞ ," and the bulk conditions for the liquid below the liquid thermal boundary layer have "o" subscripts. Interface properties have an "i" subscript.

At the interface, the energy into the interface must equal the energy out of the interface or

$$\sum_m \rho_{ma} V_{ma} h_{ma} A_i - k_a \frac{\partial T}{\partial y}|_a A_i = \sum_m \rho_{mb} V_{mb} h_{mb} A_i - k_b \frac{\partial T}{\partial y}|_b A_i \quad (III-213)$$

where

ρ_{ma} is the density of material m below the interface,

V_{ma} is the velocity component of material m below and perpendicular to the interface,

h_{ma} is the enthalpy of material m below the interface,

A_i is the surface area per unit volume of the interface,

Fig. 4.
Boundary-layer energy exchange during phase transition.

Fig. 5.
Boundary-layer mass exchange during phase transition.

k_a is the thermal conductivity of the fluid below the interface, and

T is the temperature.

Those variables with "b" subscripts on the right side of Eq. (III-213) are defined similarly for the upper side of the interface. Equation (III-213) simply states that the energy into the interface from below is equal to similar energy terms for the upper side of the interface. The first term on the left side of the equation represents the energy convected into the interface for all materials, whereas the second term accounts for heat conduction into the interface. The mass of each of these materials also must be conserved at the interface. Therefore,

$$\rho_{ma} V_{ma} A_i = \rho_{mb} V_{mb} A_i = \Gamma_m \quad . \quad (\text{III-214})$$

Also, the enthalpy difference across the interface is the heat of vaporization:

$$h_{ma} - h_{mb} = h_{g,M} \quad . \quad (\text{III-215})$$

Finally, the temperature gradients at the interface are approximated by overall heat-transfer coefficients,

$$-k_a \left. \frac{\partial T}{\partial y} \right|_a A_i = h_o A_i (T_o - T_i) \quad , \quad \text{and} \quad (\text{III-216})$$

$$-k_b \left. \frac{\partial T}{\partial y} \right|_b A_i = -h_\infty A_i (T_\infty - T_i) \quad . \quad (\text{III-217})$$

In this model, the interfacial temperature is a single temperature, whereas for the SIMMER-II simple vaporization-condensation model, the saturation temperatures can provide several interfacial temperatures. Combining Eqs. (III-213) through (III-217) yields

$$\sum_m \rho_{ma} V_{ma} A_i h_{g,M} = h_o A_i (T_i - T_o) + h_{\infty} A_i (T_i - T_{\infty}) \quad , \quad \text{or} \quad (\text{III-218})$$

$$\sum_m \Gamma_m h_{g,M} = h_o A_i (T_i - T_o) + h_{\infty} A_i (T_i - T_{\infty}) \quad . \quad (\text{III-219})$$

This equation is very similar to Eq. (III-207) of the SIMMER-II simple vaporization-condensation model except that all modes of phase transition are included on the left side of the equation. The simpler model allows only a single phase-transition rate.

The individual material boundary layer equations for the vapor side are derived without consideration of mass-transfer resistances, nonequilibrium thermodynamics, or other phenomena occurring in the liquid. External constraints between the mass-transfer rates Γ_m and the surface mass fractions x_{mi} control the liquid-side phenomena. At the interface, the mass-transfer rate of an individual material is related to the overall mixture mass-transfer rate by

$$\rho_m V_m \big|_i A_i = \Gamma_m = \rho V x_{mi} A_i - \rho D_m \frac{\partial x_m}{\partial y} \big|_i A_i \quad , \quad (\text{III-220})$$

where the total mass flux ρV is equal to the sum of the individual mass fluxes

$$\rho V = \sum_m \rho_m V_m \quad . \quad (\text{III-221})$$

The second term on the right side of Eq. (III-220) accounts for the mass diffusion of material m through the boundary layer; x_m is the mass fraction of material m , and D_m is the mass diffusivity of material m in the boundary layer. Equations (III-219) and (III-220) are rearranged into the equations

$$\Gamma_m - x_{mi} \sum_k \Gamma_k = -\rho D_m \frac{\partial x_m}{\partial y} \big|_i A_i \quad , \quad (\text{III-222})$$

whose solutions are the mass-transfer rates Γ_m .

Equation (III-222) is singular because the mass fractions x_{mi} sum to unity. This singularity arises from the over-specification of the mass-transfer problem and is overcome by replacing one of the material equations with the energy equation [Eq. (III-219)]. This substitution is done carefully because the equation replaced depends on the presence of noncondensable gases. Before this substitution is made, the relationships between the surface mass fractions x_{mi} , surface temperature T_i , and material miscibility are included to close the problem.

The material mass fractions at the interface are evaluated by assuming a constant pressure through the boundary layer to the interface. In addition, the multicomponent vapors are assumed to be in saturated thermodynamic equilibrium with the liquid (or solid) condensed phase at the interface. Furthermore, in SIMMER-II the materials are assumed to be immiscible such that each material exerts its saturation vapor pressure independently of its concentration in the condensed phase.¹⁹ Thus, for the case of no noncondensable gases, the interfacial temperature is the boiling point of the multicomponent mixture, or for an immiscible system, the eutectic temperature (the temperature for which the sum of the partial pressures equals the total pressure). Therefore, the interfacial temperature for the analytic equation of state is found by solving for T_i :

$$p_i = \sum_m p_{vMi} = \sum_m p_M^* e^{\left(\frac{-T_M^*}{T_i}\right)} \quad . \quad (\text{III-223})$$

If noncondensable gases are present, the partial pressure of the noncondensable gas at the interface is included in the total pressure. Then, the pressure p_i in Eq. (III-223) should be replaced by $p_i - p_{nc,i}$ where $p_{nc,i}$ is the partial pressure of the noncondensable gas at the interface. Unfortunately, the noncondensable partial pressure at the interface is not known a priori and is part of the solution.

The mass fraction of material m at the interface is determined from its definition and the vapor equation of state:

$$x_{mi} = \frac{\rho_{mi}}{\rho_i} = \frac{p_{Gmi} \sum_k x_{ki} (\gamma_k - 1) c_{vGM(k)}}{p_i (\gamma_m - 1) c_{vGM(m)}} , \quad (\text{III-224})$$

where the individual partial pressures are determined from the vapor pressure relation and the interfacial temperature.

Equation (III-222) is simplified further by introducing the mass-transfer coefficient g . The mass-transfer coefficient is analogous to the heat-transfer coefficient. Because the heat-transfer coefficient is defined by Eq. (III-216) and (III-217), the analogous equation for the mass-transfer coefficient is

$$-\rho D_m \frac{\partial x_m}{\partial y} \Big|_i = -g_m (x_{m\infty} - x_{mi}) , \quad (\text{III-225})$$

where

g_m is the mass-transfer coefficient of material m and

$x_{m\infty}$ and x_{mi} are the mass fractions of material m in the vapor free-stream and at the interface, respectively.

An approximate mass-transfer coefficient is derived from the heat transfer coefficient by similarity of the boundary layer equations. The simplest method is to set the Lewis number equal to unity where the Lewis number is the ratio of Schmidt number to Prandtl number. Then the mass and thermal diffusivities (D_m and α , respectively) are equated and the boundary layer equations are identical. Hence, the equivalent stagnant boundary layer thicknesses are identical. Equating the thermal and material stagnant boundary layer thicknesses yields

$$\delta_{\text{thermal}} = \frac{k}{h} = \delta_{\text{mass}} = \frac{\rho D}{g^*} , \quad (\text{III-226})$$

or, after rearranging,

$$g^* = \frac{\rho D h}{k} . \quad (\text{III-227})$$

Using the Lewis number unity simplification, ($D = \frac{k}{\rho c_p}$) gives

$$g^* = \frac{h}{c_p} \quad , \quad (\text{III-228})$$

where c_p is the mixture specific heat.

The * superscript indicates that the mass-transfer coefficient is valid only for very small mass-transfer rates. For large mass-transfer rates where the boundary conditions are not identical in the heat- and mass-transfer cases, a correction factor must be used. The correction factor multiplies g^* and yields an appropriate mass-transfer coefficient g that has been corrected for the non-zero component of velocity at the surface. The correction factor is derived from the Couette flow boundary layer approximation in which derivatives in the direction parallel to the surface are neglected because they are small in comparison with derivatives normal to the surface. Thus, the material mass-continuity equation at steady state reduces to

$$D \frac{d^2 x_m}{dy^2} - v \frac{dx_m}{dy} = 0 \quad . \quad (\text{III-229})$$

This equation is integrated twice, and the boundary conditions

$$x_m = x_{mi} \quad \text{at the interface } (y = 0) \quad \text{and}$$

$$x_m = x_{m\infty} \quad \text{in the free stream } (y = \delta) \quad (\text{III-230})$$

are used to solve for the constants of integration. The resulting solution is

$$x_m = \frac{x_{m\infty}[1 - e^{(v/D)y}] - x_{mi} [e^{(v/D)\delta} - e^{(v/D)y}]}{1 - e^{(v/D)\delta}} \quad . \quad (\text{III-231})$$

By analogy to the heat-transfer situation [Eq. (III-225)], a mass-transfer coefficient is sought that yields the diffusive component of the mass flux:

$$-\rho D \left. \frac{\partial x_m}{\partial y} \right|_{y=0} = g (x_{m\infty} - x_{mi}) = g^* \left(\frac{g}{g^*} \right) (x_{m\infty} - x_{mi}) \quad . \quad (\text{III-232})$$

The term f_g ($f_g = g/g^*$) is introduced as a correction factor that, when multiplied by g^* , yields a mass-transfer coefficient g that has been modified to account for a finite mass-transfer rate. The correction factor can be found either by solving Eq. (III-232) for f_g or by evaluating the ratio

$$f_g = \frac{\left. \frac{\partial x_m}{\partial y} \right|_{y=0}}{\lim_{V \rightarrow 0} \left. \frac{\partial x_m}{\partial y} \right|_{y=0}} \quad . \quad (\text{III-233})$$

Differentiating Eq. (III-231) and substituting it into Eq. (III-233) yields

$$f_g = \frac{\frac{V\delta}{D}}{e^{\left(\frac{V\delta}{D}\right)} - 1} \quad . \quad (\text{III-234})$$

Then, using the identities $g^* = \frac{\rho D}{\delta}$ and $\rho V A_i = \sum_m \Gamma_m = \Gamma$ gives

$$f_g = \frac{\left(\frac{\Gamma}{g^* A_i} \right)}{e^{\left(\frac{\Gamma}{g^* A_i} \right)} - 1} \quad . \quad (\text{III-235})$$

If the phase transition is vaporization (Γ positive), the correction factor is less than unity, thus decreasing the values of the heat- and mass-transfer coefficients. If the net mass flux is condensation (Γ negative), the opposite

occurs. If the net mass flux is zero, the correction factor is unity as anticipated. The correction factor also is used to correct the heat-transfer coefficient for mass-transfer effects:

$$h_{\infty} = f_g h_{\infty}^* , \quad (\text{III-236})$$

where

h_{∞} is the corrected heat-transfer coefficient and

h_{∞}^* is the heat-transfer coefficient when no mass transfer takes place.

Thermophysical property variations within the boundary layer have been found to be significant during the analysis of multicomponent phase-transition experiments. Therefore, the effect of property variations has been included in the multicomponent phase-transition model to insure that SIMMER-II will predict phase-transition rates accurately.

The thermophysical properties of interest are the thermal conductivity k , viscosity μ , mass diffusivity D , specific heat at constant pressure c_p , and density ρ . The first three properties (k , μ , and D) are calculated using kinetic theory as described in Sec. III.C. The specific heat at constant pressure is obtained from the constant-volume specific heat and the specific heat ratio. The only variation in the specific heat within the boundary layer is a result of variations in composition. The vapor density is computed from the perfect gas law and varies with changes in pressure, temperature, and composition.

The mean properties used in the multicomponent phase-transition model are obtained by evaluating the free stream and vapor-liquid interface properties separately and then using the average of the two. The mean properties obtained in this fashion differ from those obtained by picking an average temperature and composition within the boundary layer because the relationships are nonlinear.

An additional nonlinearity exists for the mass diffusivity. The Lewis number unity assumption is not needed if the mass diffusivity for each pair of species is calculated and used in Eq. (III-227) to obtain an appropriate mass-transfer coefficient. In a single-component system, the phase-transition rate is heat-transfer controlled; thus, mass-transfer coefficients are not needed to predict phase-transition rates. A two-component system needs only the binary diffusion coefficient to evaluate the mass-transfer coefficient. If three or more components exist, then an additional nonlinearity appears. This

nonlinearity results from the requirement that the sum of all diffusive fluxes be zero. This requirement can be met by defining a multicomponent mass diffusivity in accordance with the Stephan-Maxwell equations or by using the unit Lewis number assumption. The multicomponent mass diffusivity is derived in Ref. 19, where the result is

$$D_{k,g} = \frac{\rho (\Gamma_k - X_{k,i} \sum_{\lambda=1}^N \Gamma_{\lambda})}{\sum_{\lambda=1}^N \frac{1}{\rho D_{k,\lambda}} (X_{\lambda,i} \Gamma_k - X_{k,i} \Gamma_{\lambda})} . \quad (\text{III-237})$$

In summary, the equations for the multicomponent phase-transition model are the energy equation,

$$\sum_m \Gamma_m h_{g,M} = h_o A_i (T_i - T_o) + h_{\infty} A_i (T_i - T_{\infty}) ; \quad (\text{III-238})$$

the mass equations,

$$\Gamma_m - x_{mi} \sum_k \Gamma_k = -g A_i (x_{m\infty} - x_{mi}) ; \quad (\text{III-239})$$

and the definition of the correction factor,

$$f_g = \frac{\left(\frac{\Gamma}{g^* A_i} \right)}{e^{\left(\frac{\Gamma}{g^* A_i} \right) - 1}} , \quad (\text{III-240})$$

where

$$g = f_g g^* ,$$

$$h_{\infty} = f_g h_{\infty}^* , \text{ and}$$

$$g^* = \frac{h_{\infty}^*}{c_p}, \text{ if the Lewis number is unity, or}$$

$$g^* = \frac{\rho D h}{k}, \text{ if the Lewis number is not unity.}$$

e. Fission Gas Release Model. SIMMER-II includes a model for the release of fission gas from within fuel grains to the grain boundaries, subsequent release from the grain boundaries to the vapor field, and escape of the remaining gas on grain boundaries with associated fabricated fuel breakup. The release rate of intergranular fission gas is determined by Darcy's Law for gas diffusion through a porous annular medium. Further, the idea of a saturated gas concentration on grain boundaries is invoked to allow excessive intragranular gas release to occur directly to the vapor field.

The experimental basis for the intragranular release rate is the direct electrical heating experiment FGR-32²⁰ in which the release rate was

$$\Gamma_{\text{Intra}}^* = 4.47 \text{ kg/m}^3/\text{s} . \quad (\text{III-241})$$

The intragranular release rate depends on several experimental conditions, including

- (1) the fuel grain radius, r_{gr} ;
- (2) the initial intragranular fission gas density in unrestructured fuel, ρ_{s8}^0 ;
- (3) the transient heating rate in the fuel , $\frac{dT_{S1}}{dt}$;
- (4) the local temperature gradient in the fuel, $\frac{dT_{S1}}{dr}$; and
- (5) the local pressure in the fuel, p .

For the FGR-32 experiment, these conditions were

$$r_{gr} = 23 \text{ } \mu\text{m} ,$$

$$\rho_{s8}^0 = 23.6 \text{ kg/m}^3 ,$$

$$\frac{dT_{S1}}{dt} = 450 \text{ K/s} ,$$

$$\frac{dT_{S1}}{dr} = 4 \cdot 10^5 \text{ K/m} , \text{ and}$$

$$p = 10^5 \text{ Pa} .$$

In SIMMER-II, the intragranular release rate is calculated to be the experimental value [Eq. (III-241)] with appropriate adjustment factors to account for deviations from the experimental conditions; that is,

$$\Gamma_{\text{Intra}} = \Gamma_{\text{Intra}}^* f_{r_{gr}} f_{\rho_{s8}}^0 f_{\frac{dT_{S1}}{dt}} f_{\frac{dT_{S1}}{dr}} f_p , \quad (\text{III-242})$$

where

Γ_{Intra} is the intragranular fission gas release rate per unit volume of unrestructured fuel,

$f_{r_{gr}}$ is the adjustment factor to account for the effect of fuel grain size,

$f_{\rho_{s8}}^0$ is the adjustment factor to account for the effect of the initial fission gas concentration,

$f_{\frac{dT_{S1}}{dt}}$ is the adjustment factor to account for the effect of the fuel heating rate,

$f_{\frac{dT_{S1}}{dr}}$ is the adjustment factor to account for the effect of the temperature gradient in the fuel, and

f_p is the adjustment factor to account for the effect of the pressure in the fuel.

The fuel grain size is assumed to affect proportionally the release rate to the surface-area-to-volume ratio, or

$$f_{r_{gr}} = \frac{2.3 \cdot 10^{-5}}{r_{gr}} . \quad (\text{III-243})$$

The dependence of the release rate on the initial fission gas density has been approximated from results reported by Gruber.²¹ For low initial fission gas concentrations ($\rho_{s8}^0 \leq 4.37 \text{ kg/m}^3$), the fission gas release rate is assumed to be

proportional to, but weakly dependent on, the concentration. Above 4.37 kg/m^3 , bubble migration effects dominate the release rate, and the release rate increases until the initial concentration exceeds approximately 87.4 kg/m^3 . Above 87.4 kg/m^3 , bubble coalescence becomes important and decreases the dependence on the initial concentration. Using the results of Gruber as a basis, the adjustment factor for the initial fission gas density is

$$f_{\rho_{s8}}^0 = \begin{cases} 0.02174 \rho_{s8}^0, & \text{for } \rho_{s8}^0 \leq 4.37 \text{ kg/m}^3, \\ 0.04144 \rho_{s8}^0 - 0.086, & \text{for } 4.37 \text{ kg/m}^3 < \rho_{s8}^0 \leq 21.9 \text{ kg/m}^3, \\ 0.105 \rho_{s8}^0 - 1.478, & \text{for } 21.9 \text{ kg/m}^3 < \rho_{s8}^0 \leq 87.4 \text{ kg/m}^3, \\ 0.05 \rho_{s8}^0 + 3.329, & \text{for } \rho_{s8}^0 > 87.4 \text{ kg/m}^3. \end{cases} \quad (\text{III-244})$$

Gruber's results are approximated again to obtain the effect of heating rate on the intragranular fission gas release rate. The approximations used for the adjustment factor are

$$f_{\frac{dT_{S1}}{dt}} = \begin{cases} 2.61 \cdot 10^{-3} \frac{dT_{S1}}{dt} + 0.00955, & \text{for } \frac{dT_{S1}}{dt} \leq 300 \text{ K/s}, \\ 1.383 \cdot 10^{-3} \frac{dT_{S1}}{dt} + 0.3776, & \text{for } 300 \text{ K/s} \leq \frac{dT_{S1}}{dt} \leq 1000 \text{ K/s}, \\ 8.718 \cdot 10^{-4} \frac{dT_{S1}}{dt} + 0.88885, & \text{for } \frac{dT_{S1}}{dt} > 1000 \text{ K/s}. \end{cases} \quad (\text{III-245})$$

The effect of thermal gradient on intragranular fission gas release is somewhat weak. Furthermore, SIMMER-II does not compute a temperature gradient in the fuel from which an adjustment factor could be obtained. Therefore, this dependence is neglected and

$$f \frac{dT_{S1}}{dr} = 1 \quad .$$

(III-246)

Gruber's results indicate that for the adjustment factor, the effect of pressure in the fuel is given approximately by

$$f_p = \begin{cases} 1, & \text{for } p \leq 0.25 \text{ MPa} \quad , \\ 1.1976 \cdot 10^{-7} p + 0.97006, & \text{for } 0.25 \text{ MPa} < p \leq 8.6 \text{ MPa} \quad , \\ 6.637 \cdot 10^{-7} p - 3.7078, & \text{for } p > 8.6 \text{ MPa} \quad . \end{cases} \quad (\text{III-247})$$

Thus, given the fuel grain size, the initial fission gas density in unrestructured fuel, the fuel heating rate, and the internal fuel pressure, the intragranular fission gas release rate from unrestructured fuel is obtained by using these adjustment factors in Eq. (III-242). This release rate applies only when the fuel temperature is above its steady-state value; otherwise zero release is assumed.

Intergranular fission gas release is determined from the diffusion rate of fission gas along the grain boundaries in the unrestructured fuel. From Darcy's Law for gas diffusion through a porous annular region, this diffusion rate for a fuel pin of length L is given by²²

$$\frac{dm}{dt} = A_{ursf} \Gamma_{inter} = \frac{\pi K L (p_i^2 - p_o^2) \bar{p}_{sg}}{\mu_{fg} p_i \ln \left(\frac{r_o}{r_i} \right)} \quad , \quad (\text{III-248})$$

where

$A_{ursf} = \pi(r_o^2 - r_i^2)$ is the cross-sectional area of unrestructured fuel,

K is the fuel grain permeability,

r_o is the outer radius of the unrestructured fuel,

r_i is the inner radius of the unrestructured fuel,

p_i is the pressure at the inner radius of the unrestructured fuel,

p_o is the pressure at the outer radius of the unrestructured fuel,

$\bar{\rho}_{sg}$ is the intergranular fission gas density in the unrestructured fuel, and

μ_{fg} is the fission gas viscosity.

The inner radius of the unrestructured fuel can be expressed as a function of the outer radius and the fraction of the fuel that is unrestructured; that is,

$$f_{ursf} = \frac{\pi(r_o^2 - r_i^2)}{\pi r_o^2} \quad , \quad \text{or} \quad (III-249)$$

$$r_i = r_o (1 - f_{ursf})^{1/2} \quad . \quad (III-250)$$

In addition, the inner pressure is expressed as

$$p_i = p_o + \Delta p_{ursf} \quad , \quad (III-251)$$

where Δp_{ursf} is the pressure difference across the unrestructured fuel.

The fission gas viscosity is assumed to behave as

$$\mu_{fg} = 1.8 \cdot 10^{-5} \left(\frac{T_{S1}}{300} \right)^{1/2} \text{ Pa} \cdot \text{s} \quad . \quad (III-252)$$

Substituting Eqs. (III-250) through (III-252) into Eq. (III-248) yields

$$\Gamma_{\text{inter}} = \frac{2 \sqrt{300} K \Delta p_{\text{ursf}} (2p_0 + \Delta p_{\text{ursf}}) \bar{p}_{s9}}{1.8 \cdot 10^{-5} \sqrt{T_{S1}} (p_0 + \Delta p_{\text{ursf}}) r_0^2 f_{\text{ursf}} \ln(1 - f_{\text{ursf}})} \quad (\text{III-253})$$

The fission-gas release rates described in Eqs. (III-242) and (III-253) are release rates for fission gas in unrestructured fuel. To obtain the mass-transfer rates per unit volume necessary for SIMMER-II, the fission gas release rates are multiplied by the volume fraction of unrestructured fuel. In addition, as fabricated fuel fails and enters the liquid field, the intergranular fission gas is released. Intragranular gas is simply lost because no fission gas component exists in the liquid field. The fraction of fabricated fuel being transferred to the liquid field that is unrestructured fuel is assumed to be the same as for intact fuel (that is, f_{ursf}). Therefore, the percentage of fission gas released as a result of fuel failure is assumed to be the same as the percentage of fuel transferred. Thus, the equations for the two fission-gas release rates are

$$\Gamma_{s8s9} = \Gamma_{\text{intra}} \alpha_{S1} f_{\text{ursf}} \quad (\text{III-254})$$

for the intragranular fission gas, and

$$\Gamma_{s9g6} = \Gamma_{\text{inter}} \alpha_{S1} f_{\text{ursf}} + \frac{\bar{p}_{s9}}{\bar{p}_{S1}} (\Gamma_{S1L1} + \Gamma_{S1L5}) \quad (\text{III-255})$$

for the intergranular fission gas.

Three additional modifications were made to Eqs. (III-254) and (III-255). These are as follows.

1. In no case can the product of the gas release rate and the time-step size exceed the fission-gas concentration.
2. The Γ_{inter} term is based on release from fuel. The inertial resistance to fission-gas motion into the flow channel and the presence of cladding can restrict these rates. To account for these effects, the user

can specify maximum mass-transfer rates before and after cladding failure. These maximums apply to the first term in Eq. (III-255).

3. The user can specify a maximum gas concentration on grain boundaries. Once the intergranular gas obtains this concentration, further intragranular gas release is shunted directly to the vapor field.

When fission gas is released from the fuel matrix to the vapor field, it carries with it a specific internal energy. Because the fission gas resides in the fuel before its release to the vapor field, its temperature is assumed to equal the fuel temperature, and the energy of the released vapor then is found from the perfect gas law. Work is done on the vapor field during fission-gas release because of the pressure-volume work of the fission gas expansion from small, high pressure bubbles in the fuel matrix to the low pressure vapor field. This pressure-volume work is approximated by the product of the average fission-gas pressure during release, $p + 1/2 \Delta p_{\text{urdf}}$, and its change in volume fraction $(\alpha_G - \Gamma_{s9g6} \Delta t / \rho_{s9})$.

E. Neutronics and Cross-Section Models

SIMMER-II includes time-dependent transport, diffusion, and point kinetics models to allow the user to make tradeoffs between accuracy and computational speed. The transport option is included for accuracy in calculations involving significant material relocation; the neutron diffusion option is provided for use in situations where its accuracy is acceptable and its greater speed is desired. Cross sections for the transport and diffusion options incorporate temperature and background effects based on the Bondarenko formalism (self-shielding factor approach).²³ A point-kinetics option computes inexpensive neutronics feedback when space-dependent neutronics is not considered essential to the calculation.

1. Time-Dependent Neutron Transport Equation. In an LMFBR accident calculation, the time-dependent power and its spatial distribution are the desired objective of the neutronics calculation. The neutron flux must first be obtained to determine the power.

The general, linear Boltzmann transport equation for the time-dependent angular flux is given below. It has been assumed that there is no neutron upscattering, as is typical for fast reactor physics calculations, and that anisotropic scattering effects can be approximated satisfactorily by modifications to isotropic cross sections. In addition, the divergence operator has been ex-

pressed in (r, z) geometry, and it is assumed that no inhomogeneous neutron sources exist except for the decay of delayed neutron precursors.

With these assumptions, the time-dependent neutron transport equation is

$$\frac{1}{V(E)} \frac{\partial \Psi}{\partial t} + \frac{\mu}{r} \frac{\partial(r\Psi)}{\partial r} - \frac{1}{r} \frac{\partial(\xi\Psi)}{\partial \omega} + \eta \frac{\partial \Psi}{\partial z} + \Sigma_t \Psi = \frac{1}{4\pi} \int_E^\infty \Sigma_s(t, r, z, E' \rightarrow E) \Phi(t, r, z, E') dE' \\ + \frac{\chi_p(E)}{4\pi k_0} \int_0^\infty \nu_p \Sigma_f(t, r, z, E') \Phi(t, r, z, E') dE' + \frac{1}{4\pi} \sum_{d=1}^{IGD} \chi_d(E) \lambda_d C_d ,$$

where

$\Psi = \Psi(t, r, z, \vec{\Omega}, E)$ denotes the neutron flux per unit volume, per unit solid angle, per unit energy at the spatial point (r, z) with direction $\vec{\Omega}$ and energy E at time t ;

r, z, μ, ξ, η are the coordinates for (r, z) geometry as shown in Fig. 6;

$V(E)$ is the magnitude of the neutron velocity;

$\Sigma_t = \Sigma_t(t, r, z, E)$ is the macroscopic total cross section;

$\Sigma_s(t, r, z, E' \rightarrow E)$ denotes the macroscopic cross section for isotropic neutron scattering from neutron energy E' to neutron energy E ;

$\Phi(t, r, z, E)$ denotes the neutron scalar flux;

k_0 denotes the initial stationary k -effective for the reactor system;

$\chi_p(E)$ is the prompt neutron fission emission spectrum;

ν_p is the prompt neutron fission yield;

$\Sigma_f(t, r, z, E)$ is the macroscopic fission cross section;

IGD is the number of delayed neutron precursor groups;

$\chi_d(E)$ is the delayed neutron emission spectrum from delayed neutron precursor group d ;

λ_d is the decay constant for delayed neutron precursor group d ; and

Fig. 6.
Neutron transport coordinates in (r, z) geometry.

$C_d = C_d(t, r, z)$ is the delayed neutron precursor concentration for precursor group d.

Equation (III-256) is converted into a difference equation in Appendix L, where the assumptions implied by a multigroup treatment of the energy dependence and an S_N treatment of the space dependence are introduced. Further, the time dependence is treated by the quasistatic method that is developed in Appendices L and M, and the approximations used in the solution techniques are discussed.

In addition to the neutron transport equation, the equations for the delayed neutron precursor concentrations are

$$\frac{\partial C_d}{\partial t} = -\lambda_d C_d + \frac{1}{k_0} \int_0^\infty \nu_d \Sigma_f(t, r, z, E') \Phi(t, r, z, E') dE' \quad d = 1, 2, \dots, \text{IGD} ,$$

(III-257)

where ν_d is the delayed neutron yield for precursor group d . As in the transport equation, the delayed neutron precursor equations are converted into difference equations. The time, space, and energy dependencies are treated consistently with those in the transport equations. A complete treatment of the delayed neutron precursor dynamics would include precursor components in all three fluid dynamics fields. Then, volatile precursors could be released to the vapor field before fuel breakup; those precursors that remain with the fuel could be transferred to the liquid and vapor fields when fuel breaks up and vaporizes. With such a treatment, each precursor group should be characterized not only by λ_d and ν_d but also by the velocity field in which it resides. Also, because the precursors pass from one field to another (as when solid fuel melts and enters the liquid field), the precursor equations should have source and loss terms not found in the usual precursor equations. In SIMMER-II, these terms are neglected, and there is no fluid-dynamic transport of precursors.

The neutron scalar flux, Φ , found as the solution in Eq. (III-256), is used to determine the power distribution in the materials in the reactor. The specific power in each material is determined from

$$Q_{Nm} = \int_0^\infty [\alpha_c^m \Sigma_c^m(t, r, z, E') + \alpha_f^m \Sigma_f^m(t, r, z, E')] \Phi(t, r, z, E') dE' ,$$

$m = 1, \dots, \text{NNMAT}$

(III-258)

where

$Q_{Nm} = Q_{Nm}(t, r, z)$ denotes the specific power (W/kg) in material m ,

α_c^m is the energy conversion factor per neutron captured in material m,

$\Sigma_c^m(t, r, z, E')$ denotes the macroscopic capture cross section per unit density of material m,

α_f^m is the energy conversion factor per fission in material m,

$\Sigma_f^m(t, r, z, E')$ denotes the macroscopic fission cross section per unit density of material m, and

NNMAT = 5 is the number of material mixtures used to compute the neutron flux. In SIMMER-II, these are the five material components: fertile fuel, fissile fuel, stainless steel, sodium, and control.

Equation (III-258) considers only the energy production resulting from neutron capture or fission. As a consequence, the energy deposition as a result of absorption of gamma photons must be included in the neutron capture term, which leads to an error in the location of the energy deposition.

2. Time-Dependent Neutron Diffusion Model. The assumptions for the neutron diffusion equation are the same as those for the transport equation, with the exception that incorporating any anisotropic scattering effects is by means of the transport cross section. The time-dependent neutron diffusion equation modeled is

$$\frac{1}{V(E)} \frac{\partial \Phi}{\partial t} - \frac{1}{r} \frac{\partial}{\partial r} \left(D r \frac{\partial \Phi}{\partial r} \right) - \frac{\partial}{\partial z} \left(D \frac{\partial \Phi}{\partial z} \right) + \Sigma_t \Phi = \int_E^\infty \Sigma_s(t, r, z, E' \rightarrow E) \Phi(t, r, z, E') dE' \\ + \frac{\chi_p(E)}{k_0} \int_0^\infty \nu_p \Sigma_f(t, r, z, E') \Phi(t, r, z, E') dE' + \sum_{d=1}^{IGD} \chi_d(E) \lambda_d C_d \quad , \quad (III-259)$$

where $D = D(t, r, z, E) = 1/[3\Sigma_{tr}(t, r, z, E)]$ is the neutron diffusion coefficient and Σ_{tr} is the transport cross section. The delayed neutron precursor equation [Eq. (III-257)], also is used with the diffusion equation.

3. Point-Kinetics Model. A point kinetics neutronics option, referred to throughout the SIMMER-II manual as the URANUS option, permits fluid dynamics calculations without using the space-dependent neutronics capability. The simplicity of the model permits faster-running calculations, making the model useful for calculations where complex neutronics effects can be neglected. It also

is a useful option for analyzing smaller scale calculations such as single-subassembly problems or verification experiments.

The point kinetics equations used in the URANUS option are

$$\frac{dN}{dt} = (\rho - \beta) \frac{N}{\Lambda} + \sum_{d=1}^{IGD} \lambda_d c_d, \text{ and} \quad (\text{III-260})$$

$$\frac{dc_d}{dt} = \lambda_d c_d + \beta_d \frac{N}{\Lambda}, \quad d = 1, 2, \dots, IGD, \quad (\text{III-261})$$

where

$N = N(t)$ is the flux amplitude,

$\rho = \rho(t)$ is the total reactivity,

Λ is the neutron generation time,
 $\beta = \sum_{d=1}^{IGD} \beta_d$ is the total effective delayed neutron fraction,
 β_d is the effective delayed neutron fraction for the d^{th} delayed neutron precursor group, and

c_d is the adjoint-flux-weighted space and energy integral of $\chi_d C_d$.

The total reactivity is composed of several components:

$$\rho = \delta\rho_p + \delta\rho_{\text{Dop}} + \delta\rho_{\text{fert}} + \delta\rho_{\text{fiss}} + \delta\rho_{\text{st}} + \delta\rho_{\text{Na}} + \delta\rho_{\text{cont}}, \quad (\text{III-262})$$

where

$\delta\rho_p$ is the driving or programmed reactivity,

$\delta\rho_{\text{Dop}}$ is the Doppler or fuel temperature reactivity feedback,

$\delta\rho_{\text{fert}}$ is the fertile fuel displacement reactivity feedback,

$\delta\rho_{\text{fiss}}$ is the fissile fuel displacement reactivity feedback,

$\delta\rho_{\text{st}}$ is the stainless-steel displacement reactivity feedback,

$\delta\rho_{\text{NA}}$ is the sodium displacement reactivity feedback, and

$\delta\rho_{\text{cont}}$ is the control displacement reactivity feedback.

The calculation of these reactivity components is described in Chap. V.

The point-kinetics model also includes a decay heat model for which the equation is

$$\frac{dH_i}{dt} = -\lambda_{hi} H_i + \beta_{hi} N \quad , \quad (\text{III-263})$$

where

H_i is the decay heat rate for the i^{th} decay heat component,

λ_{hi} is the decay constant for the i^{th} decay heat component, and

β_{hi} is the decay heat fraction for the i^{th} decay heat component.

4. Cross-Section Model. The cross sections appearing in the multigroup versions of Eqs. (III-256) and (III-258) are determined using the Bondarenko formalism.²³ In this formalism, the capture, fission, elastic scattering, and transport cross sections for each isotope are calculated for selected temperatures by multiplying input microscopic cross sections by self-shielding factors, f , interpolated from input tables. Thus,

$$\bar{\sigma} = f\sigma \quad , \quad (\text{III-264})$$

where

$\bar{\sigma}$ is the effective self-shielded cross section and

σ is the infinitely dilute cross section.

The shielding factor depends on temperature and a background cross section (an effective shielded microscopic cross section for all isotopes in a mixture other than the isotope being considered). Because the background cross section depends on f , the process of determining cross sections is iterative.

During the time-dependent calculation, the background cross section, $\sigma_{0,i}$, is determined for each isotope i at each mesh point for each energy group as

$$\sigma_{o,i} = \frac{1}{N_i} \sum_{j \neq i} N_j \bar{\sigma}_{t,j} \quad (III-265)$$

where

N_i denotes the atom density for the i^{th} isotope and

$\bar{\sigma}_{t,j}$ denotes the shielded microscopic total cross section for the j^{th} isotope.

From the isotope temperature and background cross section, the values of the shielding factors are interpolated from the tables and the effective cross section is calculated from Eq. (III-264). The effective cross sections then are multiplied by the isotopic atom densities and summed to obtain the macroscopic cross sections used in Eqs. (III-256) and (III-259).

As an option, SIMMER-II may use a neutronics preprocessor called MXS to improve both the physics treatment (interpolation methods) and the efficiency of the cross-section shielding calculation. One of the techniques used to improve the efficiency is the combination of isotopes into materials. Some error may be introduced if, during a calculation, the composition of some cell includes materials with a common isotope. The assessment of such errors is covered in Appendix R.

REFERENCES

1. J. E. Cahalan and D. R. Ferguson, "A Preliminary User's Guide to Version 1.0 of the SAS3D LMFBR Accident Analysis Computer Code," Argonne National Laboratory unnumbered report (July 1977).
2. A. E. Waltar, W. L. Partain, D. C. Kolesar, L. P. O'Dell, A. Padilla, Jr., J. C. Sonnichsen, N. P. Wilburn, and H. J. Willenberg, "Melt-III - A Neutronics Thermal Hydraulics Computer Program for Fast Reactor Safety Analysis," Hanford Engineering Development Laboratory report HEDL-TME 74-47 (December 1974).
3. R. D. Burns III, "A Parametric Approach to LMFBR Safety Evaluation," Ph.D. Thesis, Purdue University (1976).
4. B. I. Bennett, J. D. Johnson, G. I. Kerley, and G. T. Rood, "Recent Developments in the Sesame Equation of State Library," Los Alamos Scientific Laboratory report LA-7130 (February 1978).
5. J. F. Barnes and G. P. Devault, "An Implied Temperature Dependence of the Gruneisen Parameter," Bull. A. Phys. Soc. 20, 1512 (1975).

6. J. F. Barnes and G. P. Devault, "Equations of State for LMFBR Materials," in "Proc. Int. Meeting on Fast Reactor Safety and Related Physics" ERDA report CONF-761001 (1976), Vol. IV, pp. 1744-1756.
7. Z. W. Salsburg and W. Fickett, "Mixture Theories Applied to Systems of Rigid Spheres," Los Alamos Scientific Laboratory report LA-2667 (May 1972).
8. S. Chapman and T. G. Cowling, Mathematical Theory of Nonuniform Gases (Cambridge University Press Cambridge, England, 1951), Second Edition.
9. R. B. Bird, W. E. Stewart, and E. N. Lightfoot, Transport Phenomena (John Wiley and Sons, Inc., New York, 1960), pp. 19-25.
10. R. C. Reid, J. M. Prausnitz, and T. K. Sherwood, The Properties of Gases and Liquids (McGraw-Hill, Inc., New York, 1977), pp. 394-402.
11. D. K. Edwards, V. E. Denny, and A. F. Mills, Transfer Processes (Holt, Rinehart, and Winston, Inc., New York, 1973), pp. 206-213.
12. B. W. Spencer, R. E. Henry, H. K. Fauske, G. T. Goldfuss, and L. R. Roth, "Summary and Evaluation of Reactor-Material Fuel Freezing Tests," in Proc. Int'l. Mtg. on Fast Reactor Technology (Seattle, Washington, August 19-23, 1979) (Amer. Nucl. Soc. LaGrange Park, Illinois 1979), pp. 1766-1755.
13. M. Sevik and S. H. Park, "The Splitting of Drops and Bubbles by Turbulent Fluid Flow," Fluids Engineering 95, 53-60 (March 1973).
14. M. M. El-Wakil, Nuclear Power Engineering (McGraw-Hill Book Company, Inc., New York, 1962), p. 250.
15. F. W. Dittus and L. M. K. Boelter, Heat Transfer in Automobile Radiators of the Tubular Type, (Univ. Calif. Press, Berkeley, 1930) p. 443.
16. W. M. Rohsenow and H. Y. Choi, Heat, Mass, and Momentum Transfer (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1961), p. 202.
17. Graham B. Wallis, One-Dimensional Two-Phase Flow (McGraw-Hill Book Company, New York, 1969), pp. 49-51.
18. R. W. Lockhart and R. C. Martinelli, "Proposed Correlation of Data for Isothermal Two-Phase, Two-Component Flow in Pipes," Chemical Engineering Progress 45, 39-48 (1949).
19. Olaf A. Hougen, Kenneth M. Watson, and Rolland A. Ragatz, Chemical Process Principles, Part I: Material and Energy Balances (John Wiley and Sons, Inc New York, 1956).
20. C. A. Burgess, "LMFBR Mixed-Oxide Fuels Development Semiannual Report," Hanford Engineering Development Laboratory Report HEDL-TME 75-72 (March 1976), pp. 85-116.
21. E. E. Gruber, "A Generalized Parametric Model for Transient Gas Release and Swelling in Oxide Fuels," Argonne National Laboratory report ANL-77-2 (January 1977).

22. J. R. Hofmann, "Internal Pressurization and Fission-Gas Distribution in Solid Mixed-Oxide Fuel," Argonne National Laboratory informal report ANL/RAS 76-19 (May 1976).
23. L. P. Abagyan, N. O. Bazazyants, I. I. Bondarenko, and M. N. Nikolaev, Group Constants for Nuclear Reactor Calculations (Consultants Bureau Enterprises, Inc., New York, 1964).

IV. FLUID-DYNAMICS AND STRUCTURE METHODS

The methods used to solve the equations for the fluid-dynamics and structure models are based on the techniques developed for the implicit multifield method¹ for the KACHINA computer code.² These methods have been revised to treat the numerous components required in SIMMER-II. In addition, the exchange-function models of SIMMER-II require special numerical techniques to solve the model equations, and the pressure iteration algorithm solves for the pressure corrections by matrix inversion rather than marching through the mesh and treating each cell in isolation from its neighbors.

The fluid-dynamics calculative flow is illustrated in Fig. 7. This figure shows the coupling between the fluid-dynamics calculation and the neutronics calculation. The neutronics calculation is described in greater detail in the next chapter. The fluid-dynamics calculation consists of three basic parts: (1) calculation of the exchange functions and rates; (2) explicit calculation of the continuity equations, energy equations, vapor energy equation, and portions of the momentum equations; and (3) implicit calculation of the continuity equations, momentum equations, and pressure. Areas where this flow chart is an oversimplification, such as the modification of exchange functions during the explicit calculation, will be described later. The fluid dynamics time step is estimated before the exchange-function calculation and then readjusted depending on how the calculated exchange rates are expected to affect the solution to the conservation equations.

The exchange-function calculation is spread over several subroutines. The XCHANJ subroutine controls the exchange-function calculative flow, calls the equation of state to get material temperatures, and performs the calculations of the structure-configuration model. The heat-transfer calculations are done in the TSHTR subroutine, which calls a linear equation solver to perform this implicit calculation. The vaporization and condensation models are contained in the PHASE subroutine, which iterates with the equation of state to converge that set of equations. Extra coding for the multicomponent phase-transition model is contained in the EXCHNG subroutine. The FAM subroutine completes the exchange-function calculation by computing the momentum-exchange functions, the drag heating terms, and the melting and freezing rates.

After the return from subroutine FAM, the coding for the plug model described in Sec. III.D.5.f is executed. Plug motion is updated to the beginning of the current time step based on the existing pressure field.

Fig. 7.
Coupled neutronics-fluid dynamics calculative flow.

Following a time-step update, the point kinetics equations may be solved for the relative power amplitude if the URANUS option is selected. If the URANUS option is not selected, the relative power amplitude is based on a parabolic fit over the reactivity step from coefficients determined from the space-time neutronics algorithm. If the current problem time was selected for editing the fluid-dynamics variables, that operation is performed at this point in the calculative flow so that the exchange functions can be edited if desired. Because only the internal energy distributions are stored in SIMMER-II, the equation-of-state routines are called from the edit routines to print and plot temperature distributions. Similarly, the temperatures are required for the feedback routines of the point kinetics model and for the cross-section routines of the space-time neutronics model.

The explicit part of the SIMMER-II algorithm is evaluated after the edit and is performed in subroutines EXFL1 and EXFL2 with appropriate equation-of-state calls. Here the structure and liquid energies are updated, end-of-time-step vapor conditions are estimated, tests are performed to determine whether fabricated fuel should break up or fuel crusts should no longer exist, can-wall radial motion restraints are examined, and coefficients are prepared for the pressure iteration. After these calculations are completed, a debug option is available to edit the information passed to the pressure iteration.

Subroutine IMFLUD solves the coupled continuity and momentum equations for consistent velocities, pressures, and densities. Also, because of the mixing of various vapor species with differing heat capacities and internal energies, the vapor internal energy is updated to be consistent with end-of-time-step vapor densities. If the space-time neutronics model is used, the fluid-dynamics calculation continues to cycle until the reactivity step is completed (defined in Sec. V). That is, the fluid-dynamics time steps are generally smaller than the reactivity time steps, and the fluid-dynamics problem time may be incremented several times to equal the neutronics problem time.

The problem time is divided into nonuniform time steps that are determined from time-step controls dependent on the dynamics calculated for the problem. A given time step has an interval Δt between times t^n and t^{n+1} . Variables evaluated at the beginning of a time step are indicated by a superscript n . Those at the advanced time have a superscript $n+1$. To simplify later notation, the superscript n for beginning-of-time-step variables will be omitted except in cases where it is necessary to avoid confusion.

Quasi-implicit methods often are used to estimate rate processes in SIMMER-II. These methods often estimate advanced time values for variables, but these estimates are not the final advanced time values. To distinguish these estimates from the final values, a tilde ($\sim$) is placed over the variable (for example, $\tilde{T}$). If, in addition, an iterative procedure is required to solve for final advanced time values, variables depending on the iteration number have the superscript κ to distinguish "new" values (for example, $T^{\kappa+1}$), from "old" values (for example, T^{κ}).

A. Spatial Computing Mesh

The two-dimensional geometric domain for a SIMMER-II calculation can be either a (x,z) plane uniform and infinite in the third dimension ($I_{\text{GEOM}} = 0$) or a (r,z) right circular axisymmetric cylinder ($I_{\text{GEOM}} = 1$). At present, only a point kinetics option is permitted with plane geometry. Spatially dependent neutron transport and diffusion theory are additional neutronics options available in cylindrical geometry. Plane geometry can be used to model local subassembly behavior and experimental situations, and cylindrical geometry can be used for these or full-scale reactor models. Figure 8 shows a spatial model for a typical SIMMER-II calculation in cylindrical geometry. The radial dimension of the cylinder is divided into $\bar{I}$ radial intervals with radial mesh width Δr_i in the i th interval. The radial mesh cell index i increases radially, and the value of Δr_i can differ for each i . Similarly, the axial dimension of the cylindrical domain is divided into $\bar{J}$ axial intervals with axial mesh width Δz_j in the j th interval. The axial mesh cell index j increases in the upward direction and Δz_j can also differ for each j th interval.

In plane geometry, the radial direction becomes the x direction and Δr_i becomes Δx_i . In both geometries, the intersection of radial (x) interval i and axial interval j forms mesh cell i,j with volume (area)

$$V_{ij} = (2\pi r_i)^{\xi} \Delta r_i \Delta z_j, \quad (\text{IV-1})$$

where

r_i is the radial location of the center of the mesh cell,

$\xi = 1$ for cylindrical geometry, and

$\xi = 0$ for plane geometry.

Fig. 8.
SIMMER-II fluid-dynamics computing mesh.

B. Mesh-Cell Input

When preparing input for SIMMER-II, the user obtains his input data from the results of another computer code such as SAS3D³ or from design data and drawings. The nonuniform mesh capability of SIMMER-II generally allows the calculative mesh to match the physical dimensions of the reactor components, but compromises are required if unequal subassembly expansions have occurred. The number of mesh cells can be increased to obtain more detail in designated regions of a calculation; however, the increased number of mesh cells increases the calculative effort and computer time. Furthermore, small mesh cell dimensions can lead to small time steps, further increasing the calculative effort.

Small core storage limits problems run on the CDC-7600 to 1000 mesh cells; however, this problem can be circumvented on a Cray computer by recompiling with large array sizes.

Regions having similar macroscopic densities and/or temperatures are separated into mesh-point sets. As shown in Fig. 9, the geometric domain of the mesh-point set is given by NB (the bottom-most row of cells in the mesh-point set), NT (the top-most row of cells in the mesh-point set), NL (the left-most column of cells in the mesh-point set), and NR (the right-most column of cells in the mesh-point set). Unlike SIMMER-I, the input and output for SIMMER-II do not require a shift in mesh cell indexing to account for the extra boundary cell conditions required. This shift in indexing is present within the code itself.

The variables of the fluid-dynamics model equations presented in the previous chapter are calculated for each of the mesh cells, but some quantities may be zero for some calculations or for some regions of a problem. In general, the macroscopic densities, energies, and pressure are calculated as cell-centered quantities and have integer subscripts as shown in Fig. 10. Velocities are calculated at the interfaces between the mesh cells and have half-integer subscripts to indicate their position.

Fig. 9.
SIMMER-II fluid-dynamics mesh cell.

Fig. 10.
Mesh point sets.

To simplify the notation, cell-centered spatial subscripts are omitted except in cases where they are necessary to avoid confusion. For further simplification, half-integer subscripts in the radial direction are abbreviated by λ , as in $\lambda = i + 1/2$, and half-integer subscripts in the axial direction are abbreviated by θ , as in $\theta = j + 1/2$. The use of this notation is indicated several times throughout this chapter because, although it simplifies the equations, it is not a familiar notation set.

For each mesh-point set, the user enters the macroscopic densities and temperatures for the components of the structure and liquid fields. If a macroscopic density is entered, then the temperature for the corresponding energy component should be entered. Otherwise, the temperature entered should be zero. The macroscopic densities entered should yield a total volume fraction less than one, where the remaining volume fraction is available to the vapor field. The

special definition of the liquid volume fraction from Eq. (III-2) should be recalled, and the nonflow volume fraction from the region parameters should be included when calculating the total volume fraction because of the structure field. A cell will be single-phase, with compressed liquid, if the vapor volume fraction, α_G , is less than $\alpha_0(1 - \alpha_S)$, and the cell pressure (specified by the vapor state) is greater than the sum of the liquid vapor pressures. Otherwise, a cell will be two-phase. The adjustments made to the liquid densities if $\alpha_G < \alpha_0(1 - \alpha_S)$ are given by Eqs. (IV-28) to (IV-32).

The structure- and liquid-field energy component temperatures are used to determine the corresponding structure- and liquid-field energy component specific internal energies, which are obtained from the equation of state. For the analytic equation of state, these energies are determined from Eqs. (III-31) and (III-43). Also, the component temperatures are used by Eq. (III-71) to define the microscopic liquid densities for the first three components.

The vapor state is determined in one of four ways, depending on the value of the input variable IINP. In the first method (IINP = -1), the liquid temperatures are specified, and one vapor-field component is selected through the input variable ICOMPO to be in equilibrium with its liquid-field counterpart. Because the two fuel vapors (fertile and fissile) are assumed to have the same equation of state, the variable ICOMPO refers to the first four liquid-field energy components: fuel, steel, sodium, and control, in that order. Thus, the vapor-field temperature is set equal to the temperature of liquid material ICOMPO, and the cell pressure, vapor density, and vapor energy result from the equation of state. Finally, the total component mass is conserved by subtracting the vapor mass generated from the liquid field.

For the analytic equation of state, the calculation for IINP = -1 is done as follows. First, the vapor temperature and saturation temperature for component $m = \text{ICOMPO}$ are determined from

$$T_G = T_{\text{Sat},M} = T_{Lm} \quad , \quad (\text{IV-2})$$

where $M = M(m)$ ($m = \text{ICOMPO}$) is the material number for vapor-field material component m .

This allows the cell pressure to be determined from

$$p = p_M^* \exp \left(\frac{-T_M^*}{T_{\text{Sat},M}} \right) . \quad (\text{IV-3})$$

The vapor internal energy now is evaluated from

$$e_G = c_{vSM} T_{\text{Melt},M} + h_{f,M} + c_{vLM} (T_{\text{Sat},M} - T_{\text{Melt},M}) + h_{\ell g,M} - p \Delta v_M ,$$

$$\text{if } T_{\text{Sat},M} \leq \frac{2}{3} T_{\text{Crt},M} , \quad \text{or}$$

$$e_G = c_{vSM} T_{\text{Melt},M} + h_{f,M} + c'_{vLM} \left(T_{\text{Sat},M} - \frac{2}{3} T_{\text{Crt},M} \right) \quad (\text{IV-4})$$

$$+ a_{L,M} + \frac{1}{2} h_{\ell g,M} - p \Delta v_M ,$$

$$\text{if } T_{\text{Sat},M} > \frac{2}{3} T_{\text{Crt},M} .$$

The initial microscopic density for vapor-field material component ICOMPO is obtained from

$$F(\rho_{Gm} T_G) = p - \rho_{Gm} R_M T_G = 0 . \quad (\text{IV-5})$$

This is solved using a Newton-Raphson iteration where the iteration scheme is

$$(\rho_{Gm} T_G)^{\kappa+1} = (\rho_{Gm} T_G)^{\kappa} - \frac{F^{\kappa}}{\left[\frac{dF}{d(\rho_{Gm} T_G)} \right]^{\kappa}} . \quad (\text{IV-6})$$

The derivative has the form

$$\frac{dF}{d(\rho_{Gm}T_G)} = -R_M (1 + D_M) , \quad (IV-7)$$

where

$$D_M = \frac{\rho_{Gm}T_G}{R_M} \frac{dR_M}{d(\rho_{Gm}T_G)} , \quad (IV-8)$$

is evaluated from

$$\begin{aligned} D_M &= 0 && \text{for } \rho_{Gm}T_G < \text{ROGCUT} , \\ D_M &= a_{M20} + \frac{1}{2} a_{M30} (\rho_{Gm}T_G)^{1/2} && \text{for } \text{ROGCUT} \leq \rho_{Gm}T_G \leq (\rho_{Gm}T_G)_o , \\ D_M &= \frac{1}{2} a_{M21} (\rho_{Gm}T_G)^{1/2} + a_{M31} \rho_{Gm}T_G && \text{for } (\rho_{Gm}T_G)_o < \rho_{Gm}T_G \leq (\rho_{Gm}T_G)_{Crt} , \text{ and} \\ D_M &= 0 && \text{for } \rho_{Gm}T_G > (\rho_{Gm}T_G)_{Crt} . \end{aligned} \quad (IV-9)$$

In the current version of SIMMER-II, the option to input the liquid temperature above the critical temperature with $a_{G,M}$ from Eq. (III-55) less than a large number, for example, 10^{10} , has not been programmed. To determine the new component densities, we first conserve volume:

$$\alpha_G^o + \alpha_{Lm}^o = (1 - \alpha_S - \alpha_L^i) + \bar{\rho}_{Lm}^i (1 - \alpha_o) / \rho_{Lm} , \quad (IV-10)$$

and then conserve mass,

$$\alpha_G^o \rho_{Gm} + \alpha_{Lm}^o \rho_{Lm} / (1 - \alpha_o) = \bar{\rho}_{Lm}^i , \quad (IV-11)$$

where

α_G^0 is the initial (cycle zero) vapor volume fraction,

α_{Lm}^0 is the initial volume fraction for liquid-field component m,

α_S is the structure-field total volume fraction,

α_L^i is the input liquid-field volume fraction that necessarily accounts for the remaining liquid components,

$\bar{\rho}_{Lm}^i$ is the input macroscopic density for liquid component M,

ρ_{Lm} is the microscopic density for liquid component M, and

α_0 is the fraction of the flow volume fraction below which single-phase calculations are performed.

These equations can be solved for α_G^0 as

$$\alpha_G^0 = \frac{(1 - \alpha_S - \alpha_L^i) \rho_{Lm}}{[\rho_{Lm} - \rho_{Gm} (1 - \alpha_0)]} . \quad (IV-12)$$

Hence, the initial vapor macroscopic density for component m is

$$\bar{\rho}_{Gm}^0 = \alpha_G^0 \rho_{Gm} , \quad (IV-13)$$

and the initial liquid macroscopic density becomes

$$\bar{\rho}_{Lm}^0 = \bar{\rho}_{Lm}^i - \bar{\rho}_{Gm}^0 . \quad (IV-14)$$

If the vapor- and liquid-field component specified by ICOMPO is fuel, then the densities must be determined for the fertile and fissile fuel vapor. The mass fractions of the fertile fuel and fissile fuel in the vapor field are assumed to be the same as the mass fractions in the liquid fuel. These mass fractions are

$$x_{f1L1} = \frac{\bar{\rho}_{f1}}{\bar{\rho}_{L1}} , \quad \text{and}$$

$$x_{f2L1} = \frac{\bar{\rho}_{f2}}{\bar{\rho}_{L1}} , \quad (IV-15)$$

where

x_{f1L1} is the mass fraction of fertile fuel in the liquid fuel,

x_{f2L1} is the mass fraction of fissile fuel in the liquid fuel,

$\bar{\rho}_{f1}$ and $\bar{\rho}_{f2}$ are the macroscopic densities of the liquid fertile fuel and the liquid fissile fuel density components, respectively, and

$\bar{\rho}_{L1}$ is the macroscopic density of the liquid fuel energy component.

Then, the initial fuel vapor densities are calculated as

$$\bar{\rho}_{gm} = x_{f mL1} \bar{\rho}_{G1} , \quad m = 1, 2 . \quad (IV-16)$$

For the second method (IINP = 0), the vapor-field pressure is input, and this pressure again is assigned to a single vapor-field component through the input variable ICOMPO. Then, the vapor-field energy component is calculated to have a vapor temperature equal to the saturation temperature for the input pressure. The temperature of the liquid-field energy component also is set to the same saturation temperature, thus satisfying Eq. (IV-2). For the analytic equation of state, the major difference between this option and the IINP = -1 path is that we now solve for the inverse of Eq. (IV-3) or

$$T_{\text{Sat},M} = \frac{T_M^*}{\ln \left(\frac{P_M^*}{p^i} \right)} , \quad (IV-17)$$

where p^i is the input vapor-field pressure.

For the third method (IINP = 1), the vapor state is specified by the vapor temperature and the vapor-field density component microscopic densities. Both the initial vapor energy and the pressure can be obtained by using the equation of state. For the analytic equation of state, the desired pressure is assumed to be obtained by using the gas constant at infinite dilution or

$$p_{Gm} = \rho_{Gm}^i (\gamma_M - 1) c_{vGM} T_G \quad , \quad (IV-18)$$

where

ρ_{Gm}^i is the input microscopic vapor density and

$$p = \sum_{m=1}^{N_{MAT}} p_{Gm} \quad . \quad (IV-19)$$

For two-phase cells, the actual microscopic vapor density then is determined from

$$p_{Gm} = \rho_{Gm} R_M T_G \quad (IV-20)$$

using a Newton-Raphson algorithm as was done to solve Eq. (IV-5). The vapor-field material component partial pressures determine the corresponding saturation temperatures:

$$T_{Sat,M} = \frac{T_M^*}{\ln \left(\frac{p_M^*}{p_{Gm}} \right)} \quad , \quad (IV-21)$$

from which the vapor internal energy can be evaluated by

$$e_G = \sum_{m=1}^{N_{MAT}} x_m (e_{G,M}^* + c_{vGM} T_G) \quad . \quad (IV-22)$$

Because the input density has been changed, the legitimate question to ask is whether mass conservation should be attempted. In this situation, setting ICOMPO to 1 will force such an attempt for two-phase cells. The equations that must be solved are

$$\alpha_G^0 + \sum_m \alpha_{Lm}^0 = \alpha_G^i + \sum_m \alpha_{Lm}^i \quad \text{and} \quad (\text{IV-23})$$

$$\alpha_G^0 \rho_{Gm} + \alpha_{Lm}^0 \rho_{Lm} / (1 - \alpha_0) = \alpha_G^i \rho_{Gm}^i + \alpha_{Lm}^i \rho_{Lm} / (1 - \alpha_0) \quad , \quad (\text{IV-24})$$

where α_G^i is the input vapor volume fraction before mass adjustments and the α_{Lm}^i are the input liquid field component volume fractions and where we consider the four energy components that can vaporize. Formally, Eq. (IV-23) and (IV-24) can be solved to obtain

$$\alpha_G^0 = \frac{\alpha_G^i \left(1 - \sum_{m=1}^4 \frac{\rho_{Gm}^i (1 - \alpha_0)}{\rho_{Lm}} \right)}{\left(1 - \sum_{m=1}^4 \frac{\rho_{Gm} (1 - \alpha_0)}{\rho_{Lm}} \right)} \quad . \quad (\text{IV-25})$$

Unfortunately, a negative α_{Lm}^0 may result if insufficient liquid density is available, for example, in a cell that contains sodium vapor but no liquid sodium. Therefore, Eq. (IV-24) is valid for those components where

$$\alpha_G^i (\rho_{Gm} - \rho_{Gm}^i) < \alpha_{Lm}^i \rho_{Lm} / (1 - \alpha_0) \quad . \quad (\text{IV-26})$$

This still allows errors as

$$\alpha_{Lm}^i \rho_{Lm} / (1 - \alpha_0) \geq \alpha_G^0 \rho_{Gm} - \alpha_G^i \rho_{Gm}^i$$

is not guaranteed. However, an iteration is avoided, and this restriction is

believed adequate for the IINP = 1 option. Therefore, applying Eq. (IV-25) defines α_G^0 , and then

$$\begin{aligned} \bar{p}_{Gm} &= \alpha_G^0 \rho_{Gm} \quad , \quad m = 1, \dots, 5 \quad , \quad \text{and} \\ \bar{p}_{Lm} &= \max (0, \bar{p}_{Lm}^i - \bar{p}_{Gm} \alpha_G^i \rho_{Gm}^i) \quad , \quad m = 1, \dots, 4 \quad . \end{aligned} \quad (IV-27)$$

For those cells initially having single-phase flow, the initial pressure is determined in the above manner, but the single-phase state is calculated using the single-phase equation of state. For a cell to have single-phase flow, the user must have specified sufficient structure and liquid for the vapor-field volume fraction α_G to be less than the single-phase/two-phase transition volume fraction, $\alpha_{G0} = \alpha_0 (1 - \alpha_S)$. Then the vapor pressure at the liquid temperature is calculated for each nonparticulate liquid-field energy component present in the cell. That is,

$$p_{vM(m)} = p_M^* e^{\left[\frac{-T_M^*}{(T_{Lm} - T_{Sup,M})} \right]} \quad , \quad m = 1, \dots, 4 \quad , \quad (IV-28)$$

where $T_{Sup,M}$ is the superheat of material M.

The total liquid pressure at the transition is formed from these liquid pressures:

$$\hat{p}_L = \sum_{m=1}^4 p_{vM} \quad . \quad (IV-29)$$

Equations (III-58) and (III-62) are written in terms of density components and then are solved for the liquid-field volume fraction that yields the desired cell pressure:

$$\alpha_L^0 = \alpha_{Lo} + \frac{(1 - \alpha_S)}{\bar{p}_L^i} \sum_{m=1}^{NCLR} \frac{\alpha_{Lm}^i}{c_{Lm}^2} (p^0 - \hat{p}_L) \quad , \quad (IV-30)$$

where

α_L^0 is the initial liquid-field volume fraction,

$\alpha_{Lo} = 1 - \alpha_S - \alpha_{Go}$ is the liquid-field volume fraction at the transition between single- and two-phase flow,

$\bar{\rho}_L^i$ is the input liquid-field total macroscopic density,

$\alpha_{\ell m}^i$ is the input volume fraction of liquid-field density component m,

$c_{\ell m}^2$ is the square of the sonic velocity for liquid-field density component m, and

p^0 is the initial cell pressure determined from Eq. (IV-19).

The adjusted liquid-field volume fraction is assumed to apply uniformly to all liquid-field components such that

$$\bar{\rho}_{\ell m}^0 = \frac{\alpha_L^0}{\alpha_L^i} \bar{\rho}_{\ell m}^i, \quad m = 1, \dots, NCLR, \quad (IV-31)$$

where

$\bar{\rho}_{\ell m}^0$ is the initial macroscopic density of liquid-field density component m and

$\bar{\rho}_{\ell m}^i$ is the input macroscopic density of liquid-field density component m.

The adjusted liquid-field volume fraction also leads to a new initial vapor-field volume fraction:

$$\alpha_G^0 = 1 - \alpha_S - \alpha_L^0. \quad (IV-32)$$

The initial vapor state for the single-phase cells is computed in the manner discussed in Sec. III.C.1.c.

The code user should be aware of two characteristics of the input treatment for single-phase cells. First, if the input cell pressure is less than or equal to the vapor pressure sum of Eq. (IV-29), the adjusted vapor field volume fraction given by Eq. (IV-32) will make the cell two-phase. This is a likely result

if single-phase cells are attempted with $IINP \leq 0$. Second, fission gas does not appear in the definition of the vapor state of single-phase cells. If fission gas is input into a single-phase cell, SIMMER-II will compute a spurious vapor internal energy.

The fourth method ($IINP = 2$) is similar to $IINP = 1$; however, it permits conservation of mass to be maintained with significantly less effort in two-phase cells.

In this case, the gas constant, R_M , is computed directly from $\rho_{Gm} T_G$. Thus, the pressure is computed from

$$P_{Gm} = \rho_{Gm} R_M T_G \quad (IV-33)$$

without an iteration. Because the microscopic vapor density is unchanged in two-phase cells, the liquid density is not modified. Otherwise, this option proceeds identically to the $IINP = 1$ approach.

C. Exchange-Function Methods

When the field equations (the continuity, momentum, and energy equations) are finite-differenced over the mesh cells that define the domain of the SIMMER-II calculation, the physics on a scale smaller than the mesh cell size is no longer included. The physics on this micro-scale is regained through the exchange functions coupling the field and component equations. The primary calculation of the mass-exchange rates, momentum-drag functions, and total energy exchange is driven by subroutine XCHANJ, with some minor calculations and adjustments being done in subroutines EXFL1 and EXFL2. These rates and functions are passed through arrays to the EXFLUD and IMFLUD overlays, where they are used in solving the field equations.

The sequence of the calculation driven by subroutine XCHANJ is listed in Table VIII. Initially, some geometric rearrangements may be performed. Then, an equation-of-state call obtains temperatures for the structure-, liquid-, and vapor-field energy components. The vapor temperature allows the vapor mixture thermophysical properties to be calculated; the other temperatures also are required for the heat-transfer calculation. Next, the arrangement of the structure-field energy components is determined. This arrangement is based on the initial structure-field energy components in the cell and on the subsequent loss and gain of components from melting and freezing within the cell. At the same

TABLE VIII
EXCHANGE-FUNCTION CALCULATIVE SEQUENCE

- (1) Perform any desired geometric modifications.
- (2) Call the equation-of-state routine (P4) to obtain the structure-, liquid-, and vapor-field energy component temperatures.
- (3) Determine the average properties of the vapor mixture.
- (4) Determine structure-field geometric configuration and the heat-transfer parameters.
- (5) Calculate the liquid-field droplet sizes and heat-transfer coefficients.
- (6) Calculate implicitly the structure- and liquid-field energy component heat-transfer rates.
- (7) For two-phase cells, calculate the vapor-field heat-transfer coefficients and determine the vaporization and condensation rates.
- (8) Calculate the momentum-exchange functions and determine the drag heating rates.
- (9) Determine freezing, breakup, and fission gas release rates.

time, with the use of region parameter input data, we calculate the structure-field heat-transfer flow paths, heat-transfer coefficients, and heat-transfer areas. The effect of the structure arrangement on the flow hydraulic diameter also is included.

After the structure configuration is determined, the liquid-field energy component droplet sizes are calculated and used to calculate the component heat-transfer areas and coefficients. Liquid-field energy component Reynolds and Prandtl numbers also are calculated to determine liquid-structure heat-transfer coefficients. Then, based on the type of structures available, the inferred interfacial temperatures, and the amount of material, the nonequilibrium melting/freezing model or the equilibrium model is used in an implicit heat-transfer calculation. This calculation obtains structure-structure, liquid-structure, and liquid-liquid heat-transfer rates using Gauss elimination to solve the matrix equation for the energy component temperatures.

Vaporization and condensation rates are calculated only for two-phase cells. The simple phase-transition model of SIMMER-I is the default method, but the vapor species now are treated implicitly and the equations are solved by an iterative procedure. Optionally, a user may investigate the more complex multi-

component phase-transition model to obtain the vaporization and condensation rates.

Liquid-structure and vapor-structure drag terms are evaluated for either laminar or turbulent flow and include friction factor multipliers for two-phase flow. Liquid-vapor drag is calculated using the dispersed-droplet flow correlation of SIMMER-I, but the effect of different component droplet sizes is included in the calculation. Drag heating rates are evaluated from the momentum exchange functions for all two-phase cells where some liquid exists.

The freezing and structural breakup rates first are calculated based on the imbalance of heat flow to the structural surfaces. Next, the updated liquid-field and structure-field component energies resulting from the implicit liquid/structure heat-transfer calculation, the vaporization/condensation model, and the drag heating terms are compared with the liquidus and structural breakup energies, respectively. Liquid energy deficits produce structure or solid particles. Structure energy excesses produce liquid-field components. Solid particles themselves are treated as structural components in calculating these rates. Also, a user-specified particle fraction can be calculated as associated with liquid formation for the first four structural energy components. Finally, the release rates of fission gas from the structure field are determined from the fabricated fuel heating and breakup rates.

The methods for evaluating the exchange functions vary in complexity. Those models for which the model equations are evaluated simply from the independent variables do not require detailed description of the methods. However, some exchange functions are highly sensitive, nonlinear functions of the field variables and require semi-implicit solution techniques to avoid instabilities for reasonable time steps. In addition, some of the exchange function programming is highly dependent on computer binary logic, or "IF" statements. These methods are described in the following sections.

1. Performing Desired Geometric Modifications. Two calculations that can modify a cell's geometry and/or heat flow paths are done upon first examination of a cell: the initialization of vapor density and internal energy in former nonflow volume and the initialization of the can-wall set-over model. The first calculation is done following disappearance of can wall for all cells with $F_{gp} > 1$. (The quantity F_{gp} is the 28th region parameter, GPFLG.) A description of the models involved is given in Sec. III.D.2.

The introduction of sodium vapor into a region that initially represented the interior of a blanket subassembly is straightforward, but three caveats should be mentioned. These are as follows.

(a) The inserted pressure of 0.2 MPa is approximate and will change to reflect the vapor actually in a cell during the next cycle. The added sodium vapor is introduced as possessing an infinitely dilute gas constant and only fills the former nonflow volume. It must be averaged with the remaining vapor in a cell.

(b) If a user wishes to modify the coding that replaces nonflow volume with vapor, more sophisticated formulas will be required for higher temperatures, particularly if the saturation temperature of a vapor component is above two-thirds of that component's critical temperature.

(c) After initialization, the cell in question is assigned to parameter region one. To avoid reinitialization on every subsequent time step, F_{gp} should be input as less than or equal to unity in parameter region one.

The can-wall set-over model is based on the FAIL3 and F_{gp} variables as explained in Sec. III.D.2. The index indicating can-wall set-over is ICW. If no set-over is to be performed, ICW is 0. If a can-wall is set over from the left, ICW is 1. If a can-wall is set over from the right, ICW is 2. Setting can-walls over from both left and right gives an ICW of 3. Two other important variables are

$$RIMJ = V_i/V_{i-1} \quad \text{and}$$

$$RIPJ = V_i/V_{i+1} \quad , \quad (IV-34)$$

where V_i is the volume of cell i and the j subscript is omitted. The basic idea of the can-wall set-over model is to transfer energy across cell boundaries. Because the relevant cell quantities are given in per-unit-volume terms, conservation of energy requires that energy transfers to the left be multiplied by RIMJ and energy transfers to the right be multiplied by RIPJ. The remaining ideas to be associated with can-wall set-over initialization logic are as follows: (a) the steel already calculated to be gained or lost in any can-wall that is set-over from the left is added or subtracted before set-over occurs; (b) the internal energy per unit mass assigned to the set-over can-wall for the ICW = 3 case is mass averaged based on the left and right contributions; and

(3) a variable ICWP is used to indicate if heat was transferred previously to a real can-wall from the cell on its left. If so, the heat-transfer array QS must not be zeroed to preserve this energy transfer. The algorithm for the actual calculation of the energy transfer is given in Sec. IV.C.6.

2. Determining Component Temperatures. Before the exchange functions can be evaluated, an equation-of-state call is required to evaluate the component temperatures. (The methods used in the tabular equation of state are described in Sec. III.) The simple cases for the analytic equation of state are given in Eqs. (III-50) to (III-52). The more complex cases involve solving Eq. (III-49) for the vapor temperature and solving Eq. (III-43) for T_{Lm} above $2/3 T_{Crt,M}$. Equation (III-49) can be rearranged as

$$F(T_G) = \frac{e_G - \sum_{m=1}^{NMAT} x_m e_{G,M}^*}{\sum_{m=1}^{NMAT} x_m c_{vGM}} - T_G = 0 \quad . \quad (IV-35)$$

The Newton-Raphson algorithm is

$$T_G^{\kappa+1} = T_G^{\kappa} - \frac{F^{\kappa}}{\left(\frac{dF}{dT_G}\right)^{\kappa}} \quad , \quad (IV-36)$$

where

$$\frac{dF}{dT_G} = \frac{- \sum_{m=1}^{NMAT} x_m \frac{de_{G,M}^*}{dT_G}}{\sum_{m=1}^{NMAT} x_m c_{vGM}} - 1 \quad . \quad (IV-37)$$

The explicit expressions for $e_{G,M}^*$ are

$$\begin{aligned}
 e_{G,M}^* &= e_{\text{Liq},M} + c_{\text{vLM}} (T_{\text{Sat},M} - T_{\text{Melt},M}) + h_{\text{fg},M} \\
 &\quad - p\Delta v_M - c_{\text{vGM}} T_{\text{Sat},M} \quad , \quad T_{\text{Sat},M} \leq \frac{2}{3} T_{\text{Crt},M} \quad \text{and} \\
 e_{G,M}^* &= e_{\text{Liq},M} + c'_{\text{vLM}} (T_{\text{Sat},M} - \frac{2}{3} T_{\text{Crt},M}) + a_{\text{L},M} + \frac{1}{2} h_{\text{fg},M} \\
 &\quad - p\Delta v_M - c_{\text{vGM}} T_{\text{Sat},M} \quad , \quad T_{\text{Sat},M} > \frac{2}{3} T_{\text{Crt},M} \quad . \quad (\text{IV-38})
 \end{aligned}$$

Using Eq. (III-42) for the work term expression $p\Delta v_m$,

$$\begin{aligned}
 \frac{de_{G,M}^*}{dT_G} &= \frac{dT_{\text{Sat},M}}{dT_G} \left\{ c_{\text{vLM}} - c_{\text{vGM}} - \frac{h_{\text{fg},M}}{T_M^*} \left[\xi_M \left(\frac{T_M^* - T_{\text{Sat},M}}{T_{\text{Crt},M} - T_{\text{Sat},M}} \right) + 1 \right] \right\} \quad , \\
 &\quad \text{for } T_{\text{Sat},M} \leq \frac{2}{3} T_{\text{Crt},M} \quad . \quad \text{and} \\
 \frac{de_{G,M}^*}{dT_G} &= \frac{dT_{\text{Sat},M}}{dT_G} \left\{ c'_{\text{vLM}} - c_{\text{vGM}} - \frac{h_{\text{fg},M}}{T_M^*} \left[\frac{\xi_M}{2} \left(\frac{T_M^* - 2T_{\text{Sat},M}}{T_{\text{Crt},M} - T_{\text{Sat},M}} \right) + 1 \right] \right\} \quad , \\
 &\quad \text{for } T_{\text{Sat},M} > \frac{2}{3} T_{\text{Crt},M} \quad , \quad (\text{IV-39})
 \end{aligned}$$

where

$$\begin{aligned}
 \frac{dT_{\text{Sat},M}}{dT_G} &= \frac{-T_{\text{Sat},M}}{T_G \ln \left(\frac{\rho_{\text{Gm}} R_M T_G}{P_M^*} \right)} (1 + D'_M) \quad \text{and} \quad (\text{IV-40}) \\
 D'_M &= \frac{T_G}{R_M} \frac{dR_M}{dT_G} \quad .
 \end{aligned}$$

Because the form adopted for R_M , D'_M is the same as D_M given in Eq. (IV-9) if $T_G < T_{Crt,M}$. If $T_G > T_{Crt,M}$,

$$D'_M = g_M [D_M + T_G(\ln R_{Mi} - \ln f_M)/(T_G - T_{Crt,M} + a_{G,M})] \quad , \quad (IV-41)$$

where g_M , R_{Mi} , f_M and $a_{G,M}$ are defined in Eq. (III-54) and (III-55), and D_M is given by Eq. (IV-9).

The two difficulties with this iteration are that convergence is slow initially, and if $T_{Sat,M} = T_{Crt,M}$ for all components, then

$$\frac{dF}{dT_G} = 0 \quad .$$

The first problem is avoided by storing the value of the vapor temperature obtained on the previous call to the equation of state as an additional cell variable and by using this value as an initial estimate for T_G . The second problem is avoided by assigning a minimum value of $-c_{vGM} + 10 \text{ J kg}^{-1} \text{ K}^{-1}$ to $(de_{G,M}^*/dT_G)$. This ensures that (dF/dT_G) remains sufficiently far from zero to avoid numerical difficulties.

In the T_{Lm} case, when $T_{Lm} > \frac{2}{3} T_{Crt,M}$, we define a similar function, F , by

$$F = (e_{Lm} - e_{Liq,M}) - c'_{vLM}(T_{Lm} - \frac{2}{3} T_{Crt,M}) - a_{L,M} + \frac{1}{2} h_{lg,M}^+ = 0 \quad , \quad (IV-42)$$

where $h_{lg,M}^+$ is defined following Eq. (III-43). The derivative of F is evaluated as

$$\frac{dF}{dT_{Lm}} = \frac{-\xi_M}{2(T_{Crt,M} - T_{Lm})} h_{lg,M}^+ - c'_{vLM} \quad , \quad (IV-43)$$

and the initial estimate of T_{Lm} , T_{Lm}^0 for the iteration is

$$T_{Lm}^0 = \min [T_{Crt,M} - 2 \text{ K}, \max (T_{Sat,M}, \frac{2}{3} T_{Crt,M})] \quad . \quad (IV-44)$$

Because the iteration is done only if $e_{Lm} < e_{Crt,M}$, it is only applicable when $T_{Lm} < T_{Crt,M}$. If any $T_{Lm}^{k+1} > T_{Crt,M}$ (perhaps because of a poor initial estimate), the algorithm becomes

$$T_{Lm}^{k+1} = \frac{1}{2} (T_{Lm}^k + T_{Crt,M}) \quad (IV-45)$$

until the Newton-Raphson iteration can obtain a T_{Lm}^{k+1} below $T_{Crt,M}$.

3. Assigning Control and Fuel Subassemblies to the Same Mesh Cell. The structure-configuration model assumes a mesh cell cannot contain both fuel pins and control pins simultaneously. This is because the structure geometry represents either a fuel pin subassembly or a control pin subassembly but not both. However, in an (r,z) geometry model of the reactor, both fuel and control subassemblies can exist in the same annular ring. Control subassemblies in the same radial mesh cells as driver subassemblies can be neglected, or the subassemblies can be replaced by fuel subassemblies to conserve the actual volume of the annular ring. The first approximation introduces a significant error into the criticality prediction for the reactor. Even though the system is normalized to give the initial input reactivity, the evaluated power distribution will be wrong. The second approximation gives an incorrect fuel inventory and an even worse error in power distribution.

This deficiency can be corrected by tricking SIMMER-II into including control subassembly material in a fuel subassembly mesh cell. Both fuel subassembly and control subassembly materials can be assigned to the same mesh cell by using a negative value for the region parameter NTR [RGTyp (IREG) internally]. The negative sign indicates that all control subassembly materials (including control, steel, and sodium) have been assigned to the control component in the structure field and that no heat transfer between this component and all others will be evaluated. No material is assigned to the control component in the liquid and vapor field; hence, the control subassembly materials are present but effectively static in the mesh cell. The absolute value of NTR defines the type of fuel subassembly geometry that affects the fluid-dynamics and heat-transfer analyses of all other components. The control subassembly can-wall is

assumed intact with no heat transfer between it (the structure-field control component) and the fuel subassembly composition (all other components). The only mechanism for transferring energy to the structure-field control component is through internal heat generation (for example, $\gamma_{\text{cont}} > 0$ for the URANUS model).

This option can remove most of the neutronics error. However, an effective cross-section set must be prepared for the control component to model the mixture of the control subassembly materials. An effective microscopic cross section in that set would be defined by

$$\sigma_{\text{cont}} = \frac{\bar{\rho}_{\text{cont}}^{\text{steel}} \sigma^{\text{steel}} + \bar{\rho}_{\text{cont}}^{\text{NA}} \sigma^{\text{NA}} + \bar{\rho}_{\text{cont}}^{\text{cont}} \sigma^{\text{cont}}}{\bar{\rho}_{\text{cont}}}, \quad (\text{IV-46})$$

where

$\bar{\rho}_{\text{cont}}^{\ell}$ = density of the control subassembly material ℓ smeared over the subassembly,

$\bar{\rho}_{\text{cont}} = \bar{\rho}_{\text{cont}}^{\text{NA}} + \bar{\rho}_{\text{cont}}^{\text{cont}} + \bar{\rho}_{\text{cont}}^{\text{steel}}$,

σ^{ℓ} = microscopic cross section of material ℓ in the control subassembly composition, and

ℓ can be steel, sodium, or control.

The principal neutronics deficiency of this approach results from the neglect of heat transfer between the control and fuel subassembly compositions. In an accident situation, the control subassembly composition temperature may increase faster than this approach predicts. The temperature reactivity effect will be incorrect unless the control cross-section set is modified to account for the expected higher temperatures.

One way to model the expected temperature increase in the control subassembly composition is to increase artificially its gamma heating coefficient γ_{cont} . Care must be taken to avoid introducing error in other annular mesh cell rings where control and fuel exist. It is also possible to model approximately control subassembly can-wall failure and subsequent mixing with the fuel subassembly composition. This can be done by defining an appropriate failure energy for the combined control materials. Again, care must be taken to avoid modeling the behavior of control material elsewhere in the reactor

incorrectly. These suggested artifices are described to give the user a means for modeling effects that otherwise could not be modeled by the current version of SIMMER-II.

4. Structure-Configuration Method. Although the results of the structure-configuration model are obtained by simple evaluation of the model equations, the manner in which the equations are evaluated is an involved process in computer logic. The flow diagram in Fig. 11 is included to aid in following the calculative flow. Three pointers guide the calculation flow: NTR, KCHAN, and KCHPS. In addition, there are five other pointers indicating the presence of the five structure-field energy components, and these are listed in Table IX. Two other pointers, NCHT and NCAT, are used to indicate the current structure-field energy component for which the area and other parameters are being computed. NCHT indicates the component for the pin structure, whereas NCAT indicates the component for the can-wall structure. When the structure configuration is finished, these pointers indicate the structure-field energy components adjacent to the flow fields.

The indicator NTR is set equal to the absolute value of the region parameter RGTYP of Table VI. It indicates the mesh cell type and determines the arrangement of the structure-field energy components. The mesh cell description associated with each value of NTR is listed in Table X. The value of NTR applies to the initially intact system. Thus, if the pin structure has melted away in a mesh cell initially modeling the active core, the value of NTR may be one or five. If it is one, SIMMER-II will ascertain that no pin structure exists in the mesh cell and will reset NTR to five.

KCHAN is the indicator for which frozen material to expect first: KCHAN = 1 for frozen fuel and KCHAN = 2 for frozen control. Two passes are made for each frozen material: the first (KCHPS = 1) is to determine the frozen material volume fractions and heat flow areas, and the second (KCHPS = 2) is to calculate the frozen material heat-transfer coefficient. Between the two passes, the structure surface areas, flow area, and hydraulic diameter are adjusted in the manner described in Sec. III.D.1. After the two passes are completed for the first frozen material, the value of KCHAN is switched to do the second material. The calculation for the second material completes the structure-configuration calculation.

In some situations it is possible to freeze extremely small amounts of material on the pin and can-wall structure. In these situations, the thermal response time, $\tau = (\bar{\rho}c_p)/(hA)$, of the frozen material is smaller than the fluid-

Fig. 11.
Structure-configuration model flow diagram.

Fig. 11 (cont).
Structure-configuration model flow diagram.

Fig. 11 (cont).
Structure-configuration model flow diagram.

Fig. 11 (cont).
Structure-configuration model flow diagram.

TABLE IX
STRUCTURE-FIELD ENERGY COMPONENT INDICATORS

<u>Indicator^a</u>	<u>Structure Component</u>
NF1	Fabricated Fuel Pellet
NF2	Frozen Fuel
NCL	Cladding
NCON	Control Material
NCAN	Subassembly Can-Wall

^aAn indicator value of zero means the component is not present in the cell; a value of one means the component is present.

dynamics time step. Although the implicit heat-transfer calculation described in Sec. IV.C.6 can compute the heat flows and final temperature for the frozen material adequately, the simple phase-transition model does not account for the fast response time of the structure-field energy component, and numerical instabilities can result. To avoid this problem, the energy component beneath the thin layer of frozen material is assumed to be the contact surface for the flow fields, and the thermal resistance of the frozen component is added to that for the underlying component.

5. Implicit Calculation of the Liquid Droplet Size. Explicit evaluation of liquid droplet size is unstable if based on the Weber number criterion. High liquid-vapor velocity differences cause small droplet radii that then result in the prediction of high liquid-vapor drag. High drag in the next time step decreases the liquid-vapor velocity difference and results in large droplets with low liquid-vapor drag. Low drag at the next time step increases the liquid-vapor velocity difference. To avoid this, an implicit iterative scheme has been developed to couple the Weber number criterion, the liquid-vapor drag correlation, and the momentum equations for the vapor and liquid fields.

As indicated by Eq. (III-118), the liquid droplet radius is inversely proportional to the square of the liquid-vapor velocity difference for a given critical Weber number:

TABLE X
MESH-CELL TYPE

<u>Value of Indicator NTR</u>	<u>Mesh Cell Description</u>
1	Mesh cell is in a fuel pin channel with fuel pellets initially in the mesh cell (for example, active core or blanket).
2	Mesh cell is in a control rod channel with control pellets initially in the mesh cell (for example, active control region).
3	Mesh cell is in a fuel pin channel, but fuel pellets are not present in the mesh cell (for example, fission gas plenum).
4	Mesh cell is in a control rod channel, but control pellets are not present in the mesh cell (for example, control rod plenum).
5	Mesh cell is in a fuel pin channel, but no pin structure is present in the mesh cell (for example, lower axial shield of a fuel assembly).
6	Mesh cell is in a control rod channel, but no pin structure is present in the mesh cell (for example, lower axial shield of a control assembly).
7	No structure exists in the mesh cell (for example, a sodium pool).

$$r_{pm} = \frac{\sigma_{Lm} We}{2\rho_G |\vec{V}_{GL}|^2} \quad m = 1, \dots, 4 \quad (IV-47)$$

Substituting Eq. (IV-47) into Eq. (III-168) yields

$$K_{GL} = \frac{3 \bar{p}_G}{2 \alpha_G \left[\frac{\alpha_G}{1 - \alpha_S} \right]^{\lambda-1} \left[\frac{\alpha_G}{1 - \alpha_S} - \alpha_0 \right]} \left\{ 3 \nu_G \sum_{m=5}^6 \frac{\alpha_{Lm}}{r_{pm}^2} + \frac{C_D}{4} |\vec{V}_{GL}| \sum_{m=5}^6 \frac{\alpha_{Lm}}{r_{pm}} \right\}$$

$$+ \frac{3\nu_G \rho_G^2 |\vec{v}_{GL}|^4}{\frac{1}{4} We^2} \sum_{m=1}^4 \frac{\alpha_{Lm}}{\sigma_{Lm}^2} + \frac{C_D \rho_G |\vec{v}_{GL}|^3}{2 We} \sum_{m=1}^4 \frac{\alpha_{Lm}}{\sigma_{Lm}} \} , \quad (IV-48)$$

where λ is an exponent that controls the drag as the liquid-field volume fraction increases. Four summations are present in this equation because the radii of the solid particles are assumed to be independent of the liquid-vapor velocity difference.

To solve for the $|\vec{v}_{GL}|$ that gives the particle radii in Eq. (IV-47), we use the liquid-field and vapor-field momentum equations. To make the momentum equations tractable, we drop all terms except for the pressure gradient term, which will yield a velocity dependence of $\vec{v}_{GL}$. Thus, the truncated liquid-field momentum equation is

$$\bar{\rho}_L \frac{d\vec{v}_L}{dt} = -\alpha_L \nabla p - K_{GL} (\vec{v}_L - \vec{v}_G) . \quad (IV-49)$$

Here, we have dropped the momentum-convection terms, the phase-transition momentum-exchange terms, the gravitational term, and the structure drag term. We also have assumed that the liquid-field density varies little with time.

Similar assumptions are made for the vapor-field momentum equation to yield

$$\bar{\rho}_G \frac{d\vec{v}_G}{dt} = -\alpha_G \nabla p + K_{GL} (\vec{v}_L - \vec{v}_G) . \quad (IV-50)$$

Equations (IV-49) and (IV-50) are combined to give

$$\frac{d}{dt} (\vec{v}_L - \vec{v}_G) = \left(\frac{\alpha_G}{\bar{\rho}_G} - \frac{\alpha_L}{\bar{\rho}_L} \right) \nabla p - \left(\frac{K_{GL}}{\bar{\rho}_G} + \frac{K_{GL}}{\bar{\rho}_L} \right) (\vec{v}_L - \vec{v}_G) , \quad \text{or} \quad (IV-51)$$

$$\frac{d}{dt} |\vec{V}_{GL}| = \left(\frac{\alpha_G}{\bar{\rho}_G} - \frac{\alpha_L}{\bar{\rho}_L} \right) \frac{dp}{dx} - \left(\frac{K_{GL}}{\bar{\rho}_G} + \frac{K_{GL}}{\bar{\rho}_L} \right) |\vec{V}_{GL}|, \quad (IV-52)$$

where x is the coordinate in the direction $\hat{x} = \frac{\vec{V}_L - \vec{V}_G}{|\vec{V}_{GL}|}$.

Equation (IV-52) is used to solve for an estimate of the velocity difference at the end of the time step, which then is used to calculate the droplet radii. This equation, an approximation to the combined momentum equations, is used solely to obtain the particle radius. The assumptions that are made to derive and solve the equations are not well supported, but the resulting solution to the equation provides a stable solution to the overall set of momentum equations.

The terms of Eq. (IV-52) are evaluated at the mesh cell center. To obtain the cell-center pressure gradient, an average pressure gradient for the surrounding mesh cell interfaces is used:

$$\frac{dp}{dx} \approx \frac{1}{4} \left(\frac{|p_j - p_{j+1}|}{\Delta z_{j+\frac{1}{2}}} + \frac{|p_j - p_{j-1}|}{\Delta z_{j-\frac{1}{2}}} + \frac{|p_i - p_{i+1}|}{\Delta r_{i+\frac{1}{2}}} + \frac{|p_i - p_{i-1}|}{\Delta r_{i-\frac{1}{2}}} \right). \quad (IV-53)$$

The absolute values of the pressure differences are included so that if a pressure gradient exists across the mesh cell interfaces to accelerate the two fields, it will be included in Eq. (IV-47). Without the absolute values, the pressures above and below a cell could be equal to each other but different from the cell pressure, and the resulting axial pressure gradient would be zero. As a result, no acceleration would be computed. If one of the pressure differences in Eq. (IV-53) is not expected to accelerate fluid in a certain direction, such as the pressure difference between two mesh cells separated by an intact subassembly can-wall, that pressure difference is not included in Eq. (IV-53).

Equation (IV-52) could have been derived in two ways. The above derivation results in a pressure gradient that always increases the velocity difference. The other derivation would give a pressure gradient that always decreases the velocity difference. However, the drag always decreases the velocity difference. The two derivations are equivalent if $|\vec{V}_{GL}|$ is allowed to be either posi-

tive or negative. Because our initial guess is always positive and we are seeking the maximum velocity possible, Eq. (IV-52) is used.

A Newton-Raphson technique is used to solve Eq. (IV-52). First, the vector symbol is dropped, and Eq. (IV-52) is implicitly finite differenced as

$$|v_{GL}^{n+1}| = |v_{GL}^n| + \Delta \tilde{t} \left[\left(\frac{1}{\rho_G} - \frac{1}{\rho_L} \right) \frac{dp}{dx} - \left(\frac{1}{\bar{\rho}_G} + \frac{1}{\bar{\rho}_L} \right) K_{GL} |v_{GL}^{n+1}| \right] , \quad (IV-54)$$

where $\Delta \tilde{t}$ is the estimated time step used in subroutine XCHANJ.

We make the following definitions:

$$P_1 = \Delta \tilde{t} \left(\frac{1}{\rho_G} - \frac{1}{\rho_L} \right) \frac{dp}{dx} , \quad (IV-55)$$

$$K_1 = \Delta \tilde{t} \left(\frac{1}{\bar{\rho}_G} + \frac{1}{\bar{\rho}_L} \right) \frac{18 [\bar{\rho}_G \nu_G \rho_G^2]}{\alpha_G \left[\frac{\alpha_G}{1 - \alpha_S} \right]^{\lambda-1} \left[\frac{\alpha_G}{1 - \alpha_S} - \alpha_0 \right] We^2} \sum_{m=1}^4 \frac{\alpha_{Lm}}{\sigma_{Lm}^2} , \quad (IV-56)$$

$$K_2 = \Delta \tilde{t} \left(\frac{1}{\bar{\rho}_G} + \frac{1}{\bar{\rho}_L} \right) \frac{3 [\bar{\rho}_G C_D \rho_G]}{4 \alpha_G \left[\frac{\alpha_G}{1 - \alpha_S} \right]^{\lambda-1} \left[\frac{\alpha_G}{1 - \alpha_S} - \alpha_0 \right] We} \sum_{m=1}^4 \frac{\alpha_{Lm}}{\sigma_{Lm}} , \quad (IV-57)$$

$$K_3 = \Delta \tilde{t} \left(\frac{1}{\bar{\rho}_G} + \frac{1}{\bar{\rho}_L} \right) \frac{3 [\bar{\rho}_G C_D]}{8 \alpha_G \left[\frac{\alpha_G}{1 - \alpha_S} \right]^{\lambda-1} \left[\frac{\alpha_G}{1 - \alpha_S} - \alpha_0 \right]} \sum_{m=5}^6 \frac{\alpha_{Lm}}{r_{pm}} , \quad (IV-58)$$

and

$$K_4 = \Delta \tilde{t} \left(\frac{1}{\bar{\rho}_G} + \frac{1}{\bar{\rho}_L} \right) \frac{9 \bar{\rho}_G \nu_G}{2 \alpha_G \left[\frac{\alpha_G}{1 - \alpha_S} \right]^{\lambda-1} \left[\frac{\alpha_G}{1 - \alpha_S} - \alpha_0 \right]} \sum_{m=5}^6 \frac{\alpha_{Lm}}{r_{pm}^2} . \quad (IV-59)$$

Then Eq. (IV-54) becomes

$$\begin{aligned} |v_{GL}^{n+1}| &= |v_{GL}^n| + P_1 - K_1 |v_{GL}^{n+1}|^5 \\ &\quad - K_2 |v_{GL}^{n+1}|^4 - K_3 |v_{GL}^{n+1}|^2 \\ &\quad - K_4 |v_{GL}^{n+1}| , \end{aligned} \quad (IV-60)$$

where the drag terms have been evaluated at the advanced time to give a fully implicit solution. This is necessary because of the high coupling from the drag term. To solve Eq. (IV-60) using the Newton-Raphson technique, we assume that an approximate solution $|v_{GL}^k|$ exists and we want to calculate the next iterate

$$|v_{GL}^{k+1}| = |v_{GL}^k| + \delta |v_{GL}^k| . \quad (IV-61)$$

Substitution of Eq. (IV-61) into Eq. (IV-60) and neglecting second-order terms in $\delta |v_{GL}^k|$ yields

$$\delta |v_{GL}^k| = \frac{|v_{GL}^n| - |v_{GL}^k| + P_1 - (KV)_1}{1 + (KV)_2} , \quad (IV-62)$$

where

$$(KV)_1 = K_1 (|v_{GL}^k|)^5 + K_2 (|v_{GL}^k|)^4 + K_3 (|v_{GL}^k|)^2 + K_4 |v_{GL}^k| , \text{ and} \quad (IV-63)$$

$$(KV)_2 = 5 K_1 (|v_{GL}^k|)^4 + 4 K_2 (|v_{GL}^k|)^3 + 2 K_3 (|v_{GL}^k|) + K_4 . \quad (IV-64)$$

Equations (IV-61) and (IV-62) are solved iteratively until

$$\frac{\delta |V_{GL}^k|}{|V_{GL}^k|} < 0.005$$

or until the number of iterations exceeds 50. The initial guess for the iteration is $|V_{GL}^n|$, which is calculated as

$$|V_{GL}^n| = \frac{1}{2} [(U_{Gi+\frac{1}{2}} - U_{Li+\frac{1}{2}} + U_{Gi-\frac{1}{2}} - U_{Li-\frac{1}{2}})^2 + (V_{Gj+\frac{1}{2}} - V_{Lj+\frac{1}{2}} + V_{Gj-\frac{1}{2}} - V_{Lj-\frac{1}{2}})^2]^{1/2} \quad (IV-65)$$

When the final value of $|V_{GL}^{k+1}|$ is determined, it is substituted back into Eq. (IV-47). The droplet radius determined by the critical Weber number, $r_{pm,1}$, is then the maximum of this radius, and $r_{p,min}$, the minimum droplet radius. As discussed in Sec. III.D.3, these droplet radii are compared with

- (1) the radii determined by the hydraulic diameter, $r_{pm,2}$;
- (2) the radii determined by the droplet internal pressure, $r_{pm,3}$, as limited by $r_{p,min}$;
- (3) the radii determined by the available component volume, $r_{pm,4}$, as limited by $r_{p,min}$;
- (4) the radii determined by droplet coalescence, $r_{pm,5}$; and
- (5) the maximum permitted droplet radii, $r_{p,max}$.

The droplet radii chosen are

$$r_{pm} = \min [r_{pm,1}, r_{pm,2}, r_{pm,3}, r_{pm,4}, r_{pm,5}, r_{p,max}] \quad , \quad m = 1, \dots, 4 \quad ,$$

which then are used for the droplet heat-transfer coefficients, surface areas, and the liquid-vapor momentum-exchange function.

6. Implicit Heat-Transfer Calculation. The heat-transfer rates between the structure-field energy components, between the structure-field and liquid-field energy components, and between the liquid-field energy components are determined implicitly. As shown in Sec. III.D.4, all these heat-transfer rates generally have the form

$$q_{mk} = H_{mk} (\tilde{T}_k^{n+1} - \tilde{T}_m^{n+1}) , \quad (IV-66)$$

where

q_{mk} is the heat-transfer rate per unit volume from energy component k to energy component m ,

H_{mk} is the product of the overall heat-transfer coefficient and heat-transfer area per unit volume between the two energy components, and

$\tilde{T}_k^{n+1}$ and $\tilde{T}_m^{n+1}$ are the estimated advanced-time temperatures of the two energy components.

The exception to this rule is when a structure surface temperature, T_s , is presumed to exist at the fuel or steel melting temperatures. In this case, T_s replaces $\tilde{T}_k^{n+1}$ for all liquid components in computing structure temperatures, and T_s replaces $\tilde{T}_k^{n+1}$ for the structure component in computing liquid temperatures. Also, the overall heat-transfer coefficient is not used. H_{mk} is replaced by $H_{Lm} = A_{LmSk} h_{LSm}$ as given by Eq. (III-127) for liquids under these special surface conditions, and it is replaced by $H_{Sm} = A_{Sm} h_{Sm}$ for the component representing the structure surface from Eq. (III-128). Except for these special surface conditions, temperatures in Eq. (IV-66) are evaluated at advanced times to obtain sufficient implicitness to avoid instabilities.

If all but the heat-transfer and nuclear-heating terms are neglected in the liquid and structure component-energy equations, then each energy equation has the form

$$\frac{d\bar{\rho}_m e_m}{dt} = \sum_k H_{eff,mk} (T_{eff} - \tilde{T}_m^{n+1}) + \bar{\rho}_m NQ_{Nm}, \quad (IV-67)$$

where

$T_{eff} = T_s$ if conditions in Sec. III.D.4.b are satisfied and component m is a liquid or a structure surface,

$T_{eff} = \tilde{T}_k^{n+1}$ if conditions in Sec. III.D.4.b are not satisfied, or if component m does not represent a liquid or a structure surface,

$H_{eff,mk} = H_{Lm}$ if $T_{eff} = T_s$ and component m represents a liquid component,

$H_{eff,mk} = H_{Sm}$ if $T_{eff} = T_s$ and component m represents a structure component,

$H_{eff,mk} = H_{mk}$ if $T_{eff} = \tilde{T}_k^{n+1}$,

N is the relative power amplitude,

Q_{Nm} is the normalized specific internal energy-generation rate for component m, and

the summation is over all structure- and liquid-field energy components. Because there are 5 structure-field energy components and 6 liquid-field energy components, a total of 11 equations must be solved.

To solve Eq. (IV-67), we assume that the component density does not change with time and that

$$de_m = c_{vm} dT_m, \quad (IV-68)$$

where c_{vm} is the constant-volume heat capacity of component m. After finite-differencing in time, we obtain

$$\frac{\bar{\rho}_m c_{vm}}{\Delta t} (\tilde{T}_m^{n+1} - T_m^n) = \sum_k H_{eff,mk} (T_{eff} - \tilde{T}_m^{n+1}) + \bar{\rho}_m N Q_{Nm}, \quad (IV-69)$$

or

$$(\bar{\rho}_m c_{vm} + \Delta \tilde{t} \sum_k H_{eff,mk}) \tilde{T}_m^{n+1} - \Delta \tilde{t} \sum_k H_{eff,mk} T_{eff} = \bar{\rho}_m c_{vm} T_m^n + \Delta \tilde{t} \bar{\rho}_m N Q_{Nm}. \quad (IV-70)$$

Equation (IV-70) is a symmetric matrix equation for all structure- and liquid-field energy components. The solution, $\tilde{T}_m^{n+1}$, is obtained by Gauss elimination.

For situations where a structure surface temperature is fixed at T_s , the net melting or freezing heat flux at the surface, q_{ISk} , is determined from

$$q_{ISk} = \sum_m H_{Lm} (T_s - T_{Lm}) + H_{Sk} (T_s - T_{Sk}) \quad , \quad (IV-71)$$

where m is summed over the four real liquid energy components, the subscript k represents the structure component exposing a surface to the liquid, and a positive value of q_{ISk} implies freezing (a negative value implies melting). The type of material frozen or melted is governed by whether the surface exists at the steel or fuel melting temperature. The SIMMER-II indices indicating surface conditions are IABLT and ICRUST. The possibilities are shown in Table XI. If q_{ISk} is positive, the amount of material M to be frozen in a time step is estimated from

$$\frac{(q_{ISk} \Delta t)}{[h_{f,M} + c_{vLM} (\tilde{T}_{Lm}^{n+1} - T_{Melt,M})]} \quad ,$$

TABLE XI

STRUCTURE-FIELD SURFACE TEMPERATURE INDEX VALUES

<u>Vari able</u>	<u>Value</u>	<u>Interpretation</u>
IABLT	0	No fixed pin surface temperature
IABLT	1	Pin surface temperature at $T_{Melt,Fuel}$
IABLT	2	Pin surface temperature at $T_{Melt,Steel}$
ICRUST	0	No fixed can-wall surface temperature
ICRUST	1	Can-wall surface temperature at $T_{Melt,Fuel}$
ICRUST	2	Can-wall surface temperature at $T_{Melt,Steel}$

$$\frac{(|q_{ISk}| \Delta t)}{[h_{f,M} + c_{vSM} (T_{Melt,M} - \tilde{T}_{Sk}^{n+1})]}$$

where $\tilde{T}_{Sk}^{n+1}$ is the temperature of the structural component melting, that is, that corresponding to material M. If sufficient material (which is defined as an input fraction, ROSMLT, of the appropriate cell mass) does not exist to satisfy these expressions, the cell implicit heat transfer is recalculated with $T_s = \tilde{T}_k^{n+1}$; in other words, Eq. (IV-70) is solved again with IABLT and ICRUST set to zero. A recalculation without a prescribed interface temperature also is done if the heat-transfer equations have given freezing material temperatures already below the freezing point or melting material temperatures above the melting temperature.

After the implicit heat-transfer calculation has been performed, new liquid- and structure-field component energies are estimated. For the liquid field,

$$\tilde{e}_{Lm}^{n+1} = e_{Lm}^n + c_{vLM} (\tilde{T}_{Lm}^{n+1} - T_{Lm}^n) \quad , \quad m = 1, \dots, NCLE \quad , \quad (IV-72)$$

where

$\tilde{e}_{Lm}^{n+1}$ is the estimated advanced-time liquid-field energy component specific internal energy,

e_{Lm}^n is the current liquid-field energy component specific internal energy, and

c_{vLM} is the specific heat of liquid-field energy component m.

Similarly, for the structure field,

$$\tilde{e}_{Sm}^{n+1} = e_{Sm}^n + c_{vSM} (\tilde{T}_{Sm}^{n+1} - T_{Sm}^n) \quad , \quad (IV-73)$$

where

$\tilde{e}_{Sm}^{n+1}$ is the estimated advanced-time specific internal energy of structure-field energy component m,

e_{Sm}^n is the current specific internal energy of structure-field energy component m, and

c_{vSM} is the specific heat of structure-field energy component m.

These updated energies are used to determine energy fluxes per unit volume, $\tilde{q}_{Hm}$, using the expression

$$\tilde{q}_{Hm} = \bar{\rho}_m \left(\frac{\tilde{e}_m^{n+1} - e_m^n}{\Delta t} \right) - \bar{\rho}_m NQ_{Nm}. \quad (IV-74)$$

The $\tilde{q}_{Hm}$ are updated to final q_{Hm} values after vaporization-condensation is included. Component heat-transfer rates, q_{mk} , are evaluated as needed for later determination of liquid freezing rates to various surfaces as explained by Eqs. (III-199) to (III-203).

The remaining energy transfer task in this section is to describe modifications resulting from the can-wall set-over model. If ICW is 1, $\tilde{q}_{H,S4,i-1,j}$ is updated by

$$\tilde{q}_{H,S4,i-1,j} = \tilde{q}'_{H,S4,i-1,j} + \tilde{q}_{H,S4} \frac{V_{i,j}}{V_{i-1,j}}, \quad (IV-75)$$

where

$\tilde{q}_{H,S4}$ is the can-wall energy flux from Eq. (IV-74),

$\tilde{q}'_{H,S4,i-1,j}$ is the can-wall energy flux previously computed in cell (i-1,j),

$V_{i,j}$ is the volume of cell (i,j), and

$V_{i-1,j}$ is the volume of cell (i-1,j), the cell on the left.

The quantity $\tilde{e}_{S4}^{n+1}$ is used later for determining extra can-wall melting. It is reevaluated as

(IV-78)

and

$$\frac{d(\bar{\rho}_{S4} c_{S4})_{i+1,j}}{dt} = \sum_k \frac{\bar{\rho}_{S4,i+1,j} H_{S4k}}{\bar{\rho}_{S4,i\pm 1,j}} (\tilde{T}_{k,ij}^{n+1} - \tilde{T}_{S4,i+1,j}^{n+1}) + \bar{\rho}_{S4,i+1,j} NO_{NS4,ij} \quad (IV-79)$$

Adding Eqs. (IV-78) and (IV-79) will produce the can-wall equation used in Eq. (IV-67) for the case where ICW is 3 if an advanced-time can-wall temperature is defined as

$$\tilde{T}_{S4,ij}^{n+1} = (\bar{\rho}_{S4,i-1,j} \tilde{T}_{S4,i-1,j}^{n+1} + \bar{\rho}_{S4,i+1,j} \tilde{T}_{S4,i+1,j}^{n+1}) / \bar{\rho}_{S4,i\pm 1,j} \quad (IV-80)$$

(d) Having solved Eqs. (IV-67) for the $\tilde{T}_{kij}^{n+1}$, we obtain $\tilde{T}_{S4,i-1,j}^{n+1}$ from the finite-difference representation of Eq. (IV-78), or

$$\tilde{T}_{S4,i-1,j}^{n+1} = \frac{c_{vS2} \bar{\rho}_{S4,i\pm 1,j} T_{S4,i-1,j}^n + \sum_{k \neq 4} H_{S4k} \tilde{T}_{k,ij}^{n+1} \Delta \tilde{t} + \bar{\rho}_{S4,i\pm 1,j} NO_{NS4,ij} \Delta \tilde{t}}{c_{vS2} \bar{\rho}_{S4,i\pm 1,j} + \sum_{k \neq 4} H_{S4k} \Delta \tilde{t}} \quad (IV-81)$$

A similar expression is obtained for $\tilde{T}_{S4,i+1,j}^{n+1}$ by solving Eq. (IV-79). The summations over k in Eq. (IV-81) are potentially over all 11 structure and liquid energy components, but obviously fabricated fuel, cladding, and particles cannot transfer energy to can-wall.

(e) A trial value for the can-wall energy flux to the left can-wall is

(IV-78)

and

$$\frac{d(\bar{\rho}_{S4} c_{S4})_{i+1,j}}{dt} = \sum_k \frac{\bar{\rho}_{S4,i+1,j} H_{S4k}}{\bar{\rho}_{S4,i\pm 1,j}} (\tilde{T}_{k,ij}^{n+1} - \tilde{T}_{S4,i+1,j}^{n+1}) + \bar{\rho}_{S4,i+1,j} NQ_{NS4,ij} \quad .$$

(IV-79)

Adding Eqs. (IV-78) and (IV-79) will produce the can-wall equation used in Eq. (IV-67) for the case where ICW is 3 if an advanced-time can-wall temperature is defined as

$$\tilde{T}_{S4,ij}^{n+1} = (\bar{\rho}_{S4,i-1,j} \tilde{T}_{S4,i-1,j}^{n+1} + \bar{\rho}_{S4,i+1,j} \tilde{T}_{S4,i+1,j}^{n+1}) / \bar{\rho}_{S4,i\pm 1,j} \quad . \quad (IV-80)$$

(d) Having solved Eqs. (IV-67) for the $\tilde{T}_{k,ij}^{n+1}$, we obtain $\tilde{T}_{S4,i-1,j}^{n+1}$ from the finite-difference representation of Eq. (IV-78), or

$$\tilde{T}_{S4,i-1,j}^{n+1} = \frac{c_{vS2} \bar{\rho}_{S4,i\pm 1,j} T_{S4,i-1,j}^n + \sum_{k \neq 4} H_{S4k} \tilde{T}_{k,ij}^{n+1} \Delta t + \bar{\rho}_{S4,i\pm 1,j} NQ_{NS4,ij} \Delta t}{c_{vS2} \bar{\rho}_{S4,i\pm 1,j} + \sum_{k \neq 4} H_{S4k} \Delta t} \quad .$$

(IV-81)

A similar expression is obtained for $\tilde{T}_{S4,i+1,j}^{n+1}$ by solving Eq. (IV-79). The summations over k in Eq. (IV-81) are potentially over all 11 structure and liquid energy components, but obviously fabricated fuel, cladding, and particles cannot transfer energy to can-wall.

(e) A trial value for the can-wall energy flux to the left can-wall is

$$\tilde{q}_{H,S4,i-1,j}^t = \frac{\tilde{\rho}_{S4,i-1,j}}{\tilde{\rho}_{S4,i+1,j}} \sum_{k \neq 4} H_{S4k} (\tilde{T}_{k,i,j}^{n+1} - \tilde{T}_{S4,i-1,j}^{n+1}) \quad , \quad (IV-82)$$

with a similar expression for $\tilde{q}_{H,S4,i+1,j}^t$. These are normalized to $\tilde{q}_{H,S4,i,j}$ computed from Eq. (IV-74) to guarantee energy conservation and multiplied by the appropriate volume ratios to give the final energy fluxes. Of course, the contribution to the (i-1,j) cell is added to an already existing energy flux, whereas the (i+1,j) cell energy flux must be initialized. The quantities are

$$\tilde{q}_{H,S4,i-1,j} = \tilde{q}_{H,S4,i-1,j}^t \xi_N v_{i,j}/v_{i-1,j} + \tilde{q}_{H,S4,i-1,j}' \quad , \quad (IV-83)$$

and

$$\tilde{q}_{H,S4,i+1,j} = \tilde{q}_{H,S4,i+1,j}^t \xi_N v_{i,j}/v_{i+1,j} \quad , \quad (IV-84)$$

where

$$\xi_N = \tilde{q}_{H,S4} / (\tilde{q}_{H,S4,i-1,j}^t + \tilde{q}_{H,S4,i+1,j}^t) \quad .$$

(f) The right-hand energy flux is incorporated when cell (i+1,j) is treated as in the case with an ICW of 2. The energy flux to cell (i-1,j) is used to define a $\tilde{e}_{S4}^{n+1}$ for the can-wall melting calculation. This operation is performed similarly to the case with an ICW of 1 using the formula

$$\tilde{e}_{S4}^{n+1} = \min [e_{S4,i-1,j}^n, e_{Sol,2}] + \frac{\tilde{\Delta t}}{\tilde{\rho}_{S4,i-1,j}} \tilde{q}_{H,S4,i-1,j}^t \xi_N v_{i,j}/v_{i-1,j} \quad .$$

(IV-85)

The set-over density is changed back to the case when ICW is 1 in preparation for this melting calculation. (The vaporization-condensation model uses

temporary arrays. Its energy transfers for ICW#0 are assigned as explained in Sec. IV.C.9.)

7. Methods for the Simple Vaporization-Condensation Model. The method used in SIMMER-II for determining the vaporization and condensation rates differs considerably from that used in SIMMER-I. To obtain a phase-transition rate that automatically accounts for the change in saturation properties from phase transition, the vapor components are treated implicitly. Only the liquid and structure components are treated explicitly. This also differs from the SIMMER-I technique in that the solution is obtained iteratively rather than in a single step. Compared with the SIMMER-I method, the iterative solution method provides more consistent answers and intermediate results that are easier to interpret when debugging. However, the iterative technique is more complicated to describe and the solution requires more computer time.

The equations are presented in terms of the iteration values of the vapor properties. These values approach the end-of-time-step values as the iteration converges. The heat flows for phase transition at a liquid-field energy-component interface are

$$q_{IGmLm}^{\kappa+1} = H_{GLm}^{\kappa+1} (T_{Sat,M}^{\kappa+1} - T_G^{\kappa+1}) \quad (IV-86)$$

for the heat flow to the vapor field from the interface and

$$q_{ILmGm}^{\kappa+1} = H_{LGm} (T_{Sat,M}^{\kappa+1} - \hat{T}_{Lm}) \quad (IV-87)$$

for the heat flow to the liquid-field energy component from the interface. In these equations, κ is the iteration index for heat flow, and the heat-transfer coefficients and surface areas have been combined into the single variable H .

The solution method assumes that the liquid possesses sufficient thermal inertia such that liquid temperatures are not affected significantly by vaporization or condensation in any one time step. This is not a good approximation as a component's liquid volume fraction approaches zero. Examination of implicitly formulated liquid component energy equations reveals that even for infinite liquid-side heat-transfer coefficients, the energy-transfer rate to the liquid should not be larger than

$$q_{LmGm} = \frac{\bar{\rho}_{Lm} c_{vLM}}{\Delta t} (T_{Sat,M}^{n+1} - T_{Lm}^n) \quad (IV-88)$$

Because other processes can change the liquid temperature, experience has indicated that some fraction of this maximum rate is desirable. Therefore, the maximum liquid-side heat-transfer coefficient is restricted to

$$(h_{Lm} A_{Lm})_{max} = 0.9 \frac{\bar{\rho}_{Lm} c_{vLM}}{\Delta t} \quad (IV-89)$$

The vapor-field temperature in Eq. (IV-86) is evaluated at the advanced time. The liquid-field energy-component temperature also could be evaluated implicitly, but this would complicate the solution procedure further. To account partially for the thermal response of the liquid-field energy component, the temperature used in Eq. (IV-87) depends on the change in the liquid-field energy-component temperature during the time step from other heat-transfer modes. This change is predicted by the implicit heat-transfer calculation. The basis for the choice of temperature is that the liquid-side heat-transfer coefficient is usually significantly larger than the vapor-side heat-transfer coefficient. Thus, the outcome of the comparison of the liquid-field energy-component temperature with the saturation temperature determines whether vaporization or condensation is likely to occur. Also considered in the choice is that the component temperature probably will increase during condensation and decrease during vaporization. These considerations result in the following selections for the liquid-field energy-component temperature to be used in Eq. (IV-87).

$$\begin{aligned} \hat{T}_{Lm} &= \tilde{T}_{Lm}^{n+1} \quad , \quad \text{if} \quad T_{Sat,M}^n > \tilde{T}_{Lm}^{n+1} > T_{Lm}^n \\ \text{or} \quad T_{Sat,M}^n &< \tilde{T}_{Lm}^{n+1} < T_{Lm}^n \quad , \end{aligned} \quad (IV-90)$$

$$\hat{T}_{Lm} = T_{Lm}^n \quad , \quad \text{if} \quad T_{Sat,M}^n > T_{Lm}^n > \tilde{T}_{Lm}^{n+1}$$

$$\text{or } T_{\text{Sat},M}^n < T_{\text{Lm}}^n < \tilde{T}_{\text{Lm}}^{n+1}, \quad (\text{IV-91})$$

$$\begin{aligned} \hat{T}_{\text{Lm}} &= T_{\text{Sat},M}^n, & \text{if } \tilde{T}_{\text{Lm}}^{n+1} > T_{\text{Sat},M}^n > T_{\text{Lm}}^n \\ & & \text{or } \tilde{T}_{\text{Lm}}^{n+1} < T_{\text{Sat},M}^n < T_{\text{Lm}}^n. \end{aligned} \quad (\text{IV-92})$$

In general, $\tilde{T}_{\text{Lm}}^{n+1}$ and T_{Lm}^n differ little, and the above choice has little effect compared with using the advanced-time temperature. However, when the thermal response time of the liquid-field energy component is small, this approach provides stability by anticipating the change in the liquid-field component temperatures resulting from vaporization or condensation.

The heat flows from the structure-field energy component interface are evaluated in a similar manner:

$$q_{\text{IGmSk}}^{\kappa+1} = H_{\text{Gsk}}^{\kappa+1} (T_{\text{Sat},M}^{\kappa+1} - T_{\text{G}}^{\kappa+1}) \quad (\text{IV-93})$$

for heat flow to the vapor field from the interface and

$$q_{\text{ISkGm}}^{\kappa+1} = H_{\text{SGk}} (T_{\text{Sat},M}^{\kappa+1} - \tilde{T}_{\text{Sk}}^{n+1}) \quad (\text{IV-94})$$

for the heat flow to the structure-field energy component from the interface. Again, the products of the heat-transfer coefficient and the surface area are combined. The component temperature used in Eq. (IV-94) is the advanced-time temperature because the frozen material resistances are added to the resistance for the underlying component when the thermal response of the frozen material is rapid.

To obtain the expected change in the saturation conditions during the time step, the vapor-field material component continuity equations and the vapor mixture energy equation are used in a truncated form that includes only terms from phase transitions. After differencing in time, the truncated continuity equations are

$$\rho_{Gm}^{\kappa+1} = \rho_{Gm}^n - \Delta t \tilde{\Gamma}_{GLm}^{\kappa+1}, \quad m = 1, \dots, \text{NMAT}, \quad (\text{IV-95})$$

where

$\rho_{Gm}^{\kappa+1}$ is the macroscopic density of vapor-field material component m after $\kappa+1$ iterations,

Δt is the estimated time step, and

$\Gamma_{GLm}^{\kappa+1}$ is the total mass-transfer rate from vapor-field material component m after $\kappa+1$ iterations and the fission gas (component 5) is neglected.

The total phase-transition rate is given by

$$\Gamma_{GLm}^{\kappa+1} = + \frac{1}{h_{lg,M}^{\kappa+1}} [q_{IGmLm}^{\kappa+1} + q_{ILmGm}^{\kappa+1} + \sum_{k=1}^2 (q_{IGmSk}^{\kappa+1} + q_{ISkGm}^{\kappa+1})], \quad (\text{IV-96})$$

where $h_{lg,M}^{\kappa+1}$ is the heat of vaporization for material M after $\kappa+1$ iterations.

Equations (IV-95) and (IV-96) are the first equations solved in the phase-transition model because a good estimate of the mass-transfer rate is required to solve the vapor energy equation. To evaluate the heat flows in Eq. (IV-96), $T_G^{\kappa+1}$, $H_{GLm}^{\kappa+1}$, and $H_{GSK}^{\kappa+1}$ are approximated by T_G^{κ} , H_{GLm}^{κ} , and H_{GSK}^{κ} . Then we note a strong dependence of vapor field densities on saturation conditions. Consequently, a simple evaluation of saturation properties from the equation of state and their insertion into the equations generally overestimates the degree of mass transfer. An iteration is required to couple the nonlinear relationship between $\rho_{Gm}^{\kappa+1}$ and $T_{Sat,M}^{\kappa+1}$ in the equation of state with Eq. (IV-95). The iteration proceeds by writing the differenced, truncated continuity equation for each vapor-field component as

$$\rho_{Gm}^{\kappa+1} = \rho_{Gm}^n + \frac{a_{1,m}^{\kappa} + a_{2,m}^{\kappa} T_{Sat,M}^{\kappa+1}}{h_{lg,M}^{\kappa+1}}, \quad (\text{IV-97})$$

where

$$a_{1,m}^{\kappa} = \Delta \tilde{t} (H_{GLm}^{\kappa} T_G^{\kappa} + H_{LGm} \hat{T}_{Lm}) + \Delta \tilde{t} \sum_{k=1}^2 (H_{GSk}^{\kappa} T_G^{\kappa} + H_{SGk} \tilde{T}_{Sk}^{n+1}) , \quad (IV-98)$$

and

$$a_{2,m}^{\kappa} = -\Delta \tilde{t} [H_{GLm}^{\kappa} + H_{LGm} + \sum_{k=1}^2 (H_{GSk}^{\kappa} + H_{SGk})] . \quad (IV-99)$$

To find the saturation temperature, heat of vaporization, and vapor-field component density that are consistent for satisfying Eq. (IV-97) and the equation-of-state relation between the saturation temperature and the density, the Newton-Raphson technique is used to determine the vapor-field material component density. The function used involves a rearrangement of Eq. (IV-97) as

$$F^{\hat{k}} = \alpha_G [\rho_{Gm}^n - \rho_{Gm}^{\hat{k}}] + \frac{1}{h_{\ell g,M}^{\hat{k}}} (a_{1,m}^{\kappa} + a_{2,m}^{\kappa} T_{Sat,M}^{\hat{k}}) = 0 , \quad (IV-100)$$

where $\hat{k}$ is the inner iteration index. The iteration equation then is

$$\rho_{Gm}^{\hat{k}+1} = \rho_{Gm}^{\hat{k}} - (F^{\hat{k}}) / \left(\frac{dF^{\hat{k}}}{d\rho_{Gm}^{\hat{k}}} \right) , \quad (IV-101)$$

where

$$\frac{dF^{\hat{k}}}{d\rho_{Gm}^{\hat{k}}} = \frac{1}{h_{\ell g,m}^{\hat{k}}} \left(\frac{\partial T_{Sat,M}}{\partial \rho_{Gm}} \right)^{\hat{k}} \left[a_{2,m}^{\kappa} - \frac{(a_{1,m}^{\kappa} + a_{2,m}^{\kappa} T_{Sat,m}^{\hat{k}})}{h_{\ell g,M}^{\hat{k}}} \left(\frac{\partial h_{\ell g,M}}{\partial T_{Sat,M}} \right)^{\hat{k}} \right] - \alpha_G . \quad (IV-102)$$

where the partial derivatives on the right-hand-side of Eq. (IV-102), the saturation temperature, and the heat of vaporization are obtained from the equation of state after each inner iteration. During the iteration, the vapor-field material component density is restricted to be between zero and the total liquid

and vapor mass of the material in the mesh cell. If the component density is computed to be less than zero in an iteration, the component density is set to 0.001 of the value for the previous iteration. The convergence precision achieved in this iteration is crucial for overall convergence of the model. The criteria are

- (1) A minimum of two inner iterations must be performed because consistency between $\bar{\rho}_{Gm}$ and $T_{Sat,M}$ is not guaranteed when the first $\Delta\bar{\rho}_{Gm}$ calculation is performed.
- (2) Either the saturation temperature must be converged to within 10^{-8} K, or the change in each material component $\Delta\bar{\rho}_{Gm}$, evaluated by Eq. (IV-101), must be less than δ ,

where

$$\delta = \delta_0 \max (10^{-7}, \rho_{Gm}) \frac{kg}{m^3}, \text{ except that}$$

$$\delta = 5 \times 10^{-4} \delta_0 \rho_{Gm}, \text{ if } T_{Crt,M} - 0.0625 \text{ K} < T_{Sat,M} < T_{Crt,M}, \text{ and}$$

$$\delta_0 = \text{EPHASE, an input parameter}.$$

In some cases, for example, for $T_{Sat,M}$ near $T_{Crt,M}$, the derivatives change so rapidly that convergence will not occur unless the initial estimated conditions are sufficiently close to the converged solution. Methods for these cases are discussed in Appendix E. Also, a quasi-equilibrium approximation for very large heat-transfer coefficients has been eliminated from this version of SIMMER-II because of difficulties with energy conservation.

The second set of equations solved corrects for the vapor-side heat-transfer coefficient based on the mass transfer at the various surfaces modeled by Eq. (IV-96). As shown in Sec. III.D.6.d, the corrected heat-transfer coefficient for condensation on liquid-field energy component m is

$$H_{GLm} = \frac{-\Gamma_{IGmLm} \bar{c}_{pG}}{e^{(-\Gamma_{IGmLm} \bar{c}_{pG}/H_{GLm}^*)} - 1}, \quad (IV-103)$$

where

H_{GLm} is the corrected product of the vapor-side heat-transfer coefficient and interfacial area per unit volume,

Γ_{IGmLm} is the condensation mass-transfer rate at the surface,

$\bar{c}_{pG}$ is the average constant-pressure specific heat of the vapor mixture, and

H_{GLm}^* is the product of the vapor-side heat-transfer coefficient and the interfacial area per unit volume in the absence of phase transitions.

If vaporization rather than condensation is occurring, then Γ_{IGmLm} in Eq. (IV-103) is replaced by $-\Gamma_{ILmGm}$. For the structure-field surfaces, the mass-transfer rate in Eq. (IV-103) is the total mass-transfer rate at the surface for all vapor-field components. Again, a high degree of coupling exists between the heat-transfer coefficient and the phase-transition rates, and instabilities can arise if Eq. (IV-103) is included either directly in the energy equation iteration or as part of an outer iteration. To provide stability, we use the definition of the phase-transition rate,

$$\begin{aligned} \Gamma_{IGmLm}^{\kappa+1} = & \frac{1}{h_{g,M}^{\kappa+1}} \{H_{LGm} (T_{Sat,M}^{\kappa+1} - \hat{T}_{Lm}) \\ & - \frac{\Gamma_{IGmLm}^{\kappa+1} \bar{c}_{pG}^{\kappa+1}}{[e^{(-\Gamma_{IGmLm}^{\kappa+1} \bar{c}_{pG}^{\kappa+1} / H_{GLm}^*)} - 1]} (T_{Sat,M}^{\kappa+1} - T_G^{\kappa})\} \quad , \end{aligned} \quad (IV-104)$$

which gives the additional coupling between the heat-transfer coefficient and the phase-transition rate. Equation (IV-104) is written for the condensation rate on liquid surfaces. The equation for vaporization is

$$\begin{aligned} \Gamma_{ILmGm}^{\kappa+1} = & \frac{1}{h_{g,M}^{\kappa+1}} \{H_{LGm} (T_{Sat,M}^{\kappa+1} - \hat{T}_{Lm}) \\ & + \frac{\Gamma_{ILmGm}^{\kappa+1} \bar{c}_{pG}^{\kappa+1}}{[e^{(\Gamma_{ILmGm}^{\kappa+1} \bar{c}_{pG}^{\kappa+1} / H_{GLm}^*)} - 1]} (T_{Sat,M}^{\kappa+1} - T_G^{\kappa})\} \quad . \end{aligned} \quad (IV-105)$$

A similar procedure is used for structure surfaces. A Newton-Raphson procedure is used to solve Eqs. (IV-104) and (IV-105) for the phase-transition rate, taking the values $\Gamma_{\text{Sat},M}^{\kappa+1}$, $\bar{c}_{pG}^{\kappa+1}$, and $h_{lg,M}^{\kappa+1}$ from the converged solution of Eq. (IV-101). Consequently, the condensation rate, $(\Gamma_{GmLm}^{\kappa+1})_{\text{ISk}}$, or the vaporization rate, $(\Gamma_{LmGm}^{\kappa+1})_{\text{ISk}}$, furnishes the single iterative variable. Details of the procedure [in other words, when to take the equivalent of Eq. (IV-104), when to use the equivalent of Eq. (IV-105), and what prescription to follow if no solution exists] are in Appendix E. After a rate is obtained, the corrected heat-transfer coefficient is evaluated from the equivalent of Eq. (IV-103) or its vaporization analogy.

Finally, the vapor energy equation is solved. The truncated vapor-mixture energy equation is

$$\begin{aligned} \bar{p}_G^{\kappa+1} e_G^{\kappa+1} = & \bar{p}_G^n e_G^n - \Delta \tilde{t} \sum_{m=1}^{\text{NMAT}} \Gamma_{GmLm}^{\kappa+1} (e_{\text{Con},M}^{\kappa+1} + h_{lg,M}^{\kappa+1}) \\ & + \Delta \tilde{t} \sum_{m=1}^{\text{NMAT}} (q_{IGmLm}^{\kappa+1} + \sum_{k=1}^2 q_{IGmSk}^{\kappa+1}) , \end{aligned} \quad (\text{IV-106})$$

where

$$\bar{p}_G^{\kappa+1} = \sum_{m=1}^{\text{NMAT}} \bar{p}_{Gm}^{\kappa+1} \quad \text{is the total macroscopic density of the vapor-field components after evaluating Eq. (IV-95),}$$

$$e_G^{\kappa+1} \quad \text{is the specific internal energy of the vapor mixture for which the equation is solved, and}$$

$$e_{\text{Con},M}^{\kappa+1} \quad \text{is the specific internal energy of liquid-field material component } m \text{ at the saturation temperature determined in evaluating Eq. (IV-95).}$$

Only the energy-transfer terms from mass transfer and the associated heat transfer are included in Eq. (IV-106). The phase transition is assumed to occur at the liquid-field and structure-field energy component interfaces, which are assumed to be at the saturation temperature. Therefore, the mass transferred from the vapor field is assumed to leave the field at saturated conditions.

Multiplication of Eq. (IV-95) by $e_G^{\kappa+1}$, summation over m , and subtraction from Eq. (IV-106) yields

$$\begin{aligned}
\bar{\rho}_G^n e_G^{\kappa+1} = \bar{\rho}_G^n e_G^n - \Delta \tilde{t} \sum_{m=1}^{N_{MAT}} \Gamma_{GLm}^{\kappa+1} (e_{Con,M}^{\kappa+1} + h_{g,M}^{\kappa+1} - e_G^{\kappa+1}) \\
+ \Delta \tilde{t} \sum_{m=1}^{N_{MAT}} (q_{IGmLm}^{\kappa+1} + \sum_{k=1}^2 q_{IGmSk}^{\kappa+1}) .
\end{aligned} \tag{IV-107}$$

Equation (IV-107) now is modified as follows.

- (1) Special energy-transport terms are not written for the fission gas component. Except for a special case of direct structure-to-vapor heat transport described in Appendix E, fission gas merely adds to the vapor density and heat capacity and modifies the vapor thermophysical properties (and consequently influences the vapor-side heat-transfer coefficients).
- (2) The case when a liquid component can vaporize totally is handled by inserting terms that place the liquid energy directly into the vapor field.
- (3) We wish to solve Eq. (IV-107) for $T_G^{\kappa+1}$ and $e_G^{\kappa+1}$. We approximate $e_G^{\kappa+1}$ by

$$e_G^{\kappa+1} = e_G^\kappa + \bar{c}_{vG}^{\kappa+1} (T_G^{\kappa+1} - T_G^\kappa) . \tag{IV-108}$$

This yields

$$\begin{aligned}
\bar{\rho}_G^n [e_G^\kappa + \bar{c}_{vG}^{\kappa+1} (T_G^{\kappa+1} - T_G^\kappa)] = \bar{\rho}_G^n e_G^n - \Delta \tilde{t} \sum_{m=1}^{N_{MAT}-1} x_m \{ \Gamma_{GLm}^{\kappa+1} [e_{Con,M}^{\kappa+1} + h_{g,M}^{\kappa+1} - e_G^\kappa \\
- \bar{c}_{vG}^{\kappa+1} (T_G^{\kappa+1} - T_G^\kappa)] - (q_{IGmLm}^{\kappa+1} + \sum_{k=1}^2 q_{IGmSk}^{\kappa+1}) \} \\
+ \sum_{m=1}^{N_{MAT}-1} (1 - x_m) \{ \bar{\rho}_{Lm}^n [\bar{e}_{Lm}^{\kappa+1} - e_G^\kappa - \bar{c}_{vG}^{\kappa+1} (T_G^{\kappa+1} - T_G^\kappa)] \} ,
\end{aligned} \tag{IV-109}$$

where

$x_m = 1$ is the nominal case and

$x_m = 0$ when all of $\bar{\rho}_{Lm}^n$ vaporizes.

To place Eq. (IV-109) into a manageable format, three dummy variables (a_m , b_m , and d_m) are defined as

$$a_m = -\Delta t \left[H_{LGm} (T_{Sat,M}^{\kappa+1} - \hat{T}_{Lm}) + \sum_{k=1}^2 H_{SGk} (T_{Sat,M}^{\kappa+1} - \tilde{T}_{Sk}^{n+1}) \right] , \quad (IV-110)$$

$$b_m = -\Delta t \left(H_{GLm}^{\kappa+1} + \sum_{k=1}^2 H_{GSk}^{\kappa+1} \right) , \text{ and} \quad (IV-111)$$

$$d_m = e_{Con,M}^{\kappa+1} + h_{Lg,M}^{\kappa+1} - e_G^{\kappa} + \bar{c}_{vG}^{\kappa+1} T_G^{\kappa} . \quad (IV-112)$$

The quantities a_m , b_m , and d_m are regarded as constants during the solution of the vapor energy equation. They use the updated saturation quantities from the first Newton-Raphson iteration and updated vapor-side heat transfer coefficients from the second iteration. Then it is possible to show that

$$\begin{aligned} & \left(\sum_{m=1}^{NMAT-1} x_m \frac{b_m \bar{c}_{vG}^{\kappa+1}}{h_{Lg,M}^{\kappa+1}} \right) (T_G^{\kappa+1})^2 - [\bar{\rho}_G^n \bar{c}_{vG}^{\kappa+1} \\ & + \sum_{m=1}^{NMAT-1} x_m \left(-b_m + \frac{a_m \bar{c}_{vG}^{\kappa+1} + b_m d_m + b_m T_{Sat,M}^{\kappa+1} \bar{c}_{vG}^{\kappa+1}}{h_{Lg,M}^{\kappa+1}} \right) \\ & + \sum_{m=1}^{NMAT-1} \bar{c}_{vG}^{\kappa+1} \bar{\rho}_{Lm}^n (1 - x_m)] T_G^{\kappa+1} \\ & + \{ -\bar{\rho}_G^n (e_G^{\kappa} - e_G^n - \bar{c}_{vG}^{\kappa+1} T_G^{\kappa}) \end{aligned}$$

$$\begin{aligned}
& - \sum_{m=1}^{NMAT-1} x_m \left[b_m T_{Sat,M}^{\kappa+1} - \frac{a_m d_m + b_m T_{Sat,M}^{\kappa+1}}{h_{g,M}^{\kappa+1}} \right] \\
& + \sum_{m=1}^{NMAT-1} \left[\bar{p}_{Lm}^M (\bar{e}_{Lm}^{n+1} - e_G^{\kappa} + \bar{c}_{vG}^{\kappa+1} T_G^{\kappa}) \right] [1 - x_m] \} \\
& = 0 \quad .
\end{aligned} \tag{IV-113}$$

This is a quadratic of the form

$$AT_G^2 + BT_G + C = 0 \tag{IV-114}$$

that is solved using the quadratic formula, assuming $T_G^{\kappa+1}$ is the only unknown. The quantity $A < 0$ by definition, so the negative square root gives the higher temperature and, in most cases, furnishes the correct solution. The exceptions and the procedure followed if $B^2 - 4AC$ is less than zero are discussed in Appendix E. The one major special case requiring immediate discussion is where we have a large vapor-side heat-transfer coefficient. In this case, the vapor continuity and vapor energy equations are coupled strongly. Experience has indicated that the best approach to this problem is to evaluate $\Delta t \Gamma_{GLm}^{\kappa+1}$ directly by using the previous expression for $\bar{p}_{Gm}^n - \bar{p}_{Gm}^{\kappa+1}$. This reduces Eq. (IV-113) to a linear equation in $T_G^{\kappa+1}$. The criterion for this case is that

$$-x_m \frac{b_m \bar{c}_{vG}^{\kappa+1}}{h_{g,m}^{\kappa+1}} > 100 \tag{IV-115}$$

for any component m .

Three special situations should be mentioned regarding the overall iteration. First, if all the liquid of a given component vaporizes, the corresponding saturation temperature returned from the model is increased such that $\Gamma_{GLm}^{\kappa+1}$ still can be calculated correctly from temperature differences. (This artificial saturation temperature appears on the SIMMER-II full print.) Second,

convergence of the model can occur either with the variables oscillating about an apparent converged value from iteration to iteration or with the variables changing monotonically in a given algebraic direction. Convergence is accelerated in the oscillatory case by Steffensen's method,⁴ where for a given variable, y :

$$y^{k+1} = y^k - \frac{(y^k - y^{k-1})^2}{y^k - 2y^{k-1} + y^{k-2}} \quad . \quad (IV-116)$$

Third, if complete convergence does not occur, $T_{Sat,M}^k$ is set to $-a_{1,m}^k/a_{2,m}^k$ and a message is printed. This error exit eliminates vaporization/condensation for such a cell for the time step in question. Generally, this warning message indicates a problem; the input and/or coding should be investigated if excessive warning messages are given.

In summary, the vaporization-condensation model is solved by performing three major computational operations successively. There is an iteration scheme to obtain phase-transition rates consistent with the saturation temperatures. A second set of iterations adjusts these rates for consistency with the vapor-side heat-transfer coefficients. The third computation takes these saturation temperatures and heat-transfer coefficients and updates the vapor temperature using the vapor energy equations. These operations are repeated until convergence is obtained. The steps in the algorithm can be related to the equations as follows.

- (1) Estimate the liquid-field energy-component temperatures from the beginning-of-time-step temperatures, the advanced-time temperatures, or the saturation temperatures using Eqs. (IV-90) through (IV-92).
- (2) Perform the inner iteration between the saturation temperatures and the vapor-field material components continuity equations using Eq. (IV-101) and the equation of state.
- (3) If the outer iteration is converged, evaluate the converged heat fluxes and phase-transition rates and exit from the phase-transition calculation.
- (4) Perform the inner iteration between the corrected vapor-side heat-transfer coefficients and the phase-transition-rate definition using Eqs. (IV-104) and (IV-105) and the associated equations for structure surfaces if required.

(5) Solve the vapor-field energy equation for the new vapor-field temperature using Eq. (IV-113).

(6) Return to Step 2.

Step (5) is performed at least once. The convergence criteria required in Step (3) are that the mass-transfer fractional change between successive outer iterations associated with H_{GLm}^{k+1} must be converged to within an input parameter (EVAPOR) and that the vapor-side heat-transfer coefficients in Step (4) must converge within 0.1% between successive outer iterations.

The vaporization-condensation model has many special cases that arise when a small amount of a liquid-field energy component is present or when the liquid-field energy-component temperatures approach or exceed the critical temperature for the material. These special cases and their treatment are detailed in Appendix E.

8. Solution Method for the Multicomponent, Phase-Transition Model. The numerical techniques for the multicomponent simultaneous heat- and mass-transfer model also are iterative because the correction factor for the vapor-side heat-transfer coefficient depends nonlinearly on the mass-transfer rate and because nonlinear relationships exist between the interfacial temperature and the interfacial vapor-field mass fractions in cases with noncondensable gases. The solution procedure is presented in a stepwise fashion, and each step can contain a set of parallel or sequential operations because of the number of possibilities.

Step 1. All the variables that do not change during the iteration are initialized. These include the free-stream mass fractions x_m , a free-stream specific heat (weighted by the mass fractions), and the heat and mass-transfer coefficients h^* and g^* .

Step 2. The presence of noncondensable gases must be ascertained because the solution technique differs when they are present.

Case 1. Noncondensable gases are not present. In this case, the interfacial temperature T_i is known to be the eutectic temperature of the multicomponent mixture that is calculated from Eq. (III-223). The interfacial temperature then is used in the vapor-pressure relationship to compute the partial pressures at the interface. From Eq. (III-224), the partial pressures determine the corresponding interfacial mass fractions x_{mi} .

Case 2. Noncondensable gases are present. In this case, the interfacial temperature is not known; therefore, an initial guess is made from which the corresponding surface mass fractions (x_{mi}) are calculated.

Step 3. The matrix equation is solved. As stated previously, Eq. (III-222) is singular unless one of the mass-transfer equations, which is determined by the presence of noncondensable gases, is replaced by the energy equation.

Case 1. Noncondensable gases are not present. For this case, good results are obtained by replacing the mass-transfer equation for the most volatile material with the energy equation. With the numerical subscripts indicating materials of increasing volatility, the matrix equation becomes the following.

$$\begin{bmatrix} (1-x_{1i}) & -x_{1i} & -x_{1i} & . & . \\ -x_{2i} & (1-x_{2i}) & -x_{2i} & . & . \\ -x_{3i} & -x_{3i} & (1-x_{3i}) & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ h_{\ell g,1} & h_{\ell g,2} & h_{\ell g,3} & . & . \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ . \\ \Gamma_N \end{bmatrix} = \begin{bmatrix} -gA_i (x_{1\infty} - x_{1i}) \\ -gA_i (x_{2\infty} - x_{2i}) \\ -gA_i (x_{3\infty} - x_{3i}) \\ . \\ . \\ h_o A_i (T_i - T_o) \\ + h_{\infty} A_i (T_i - T_{\infty}) \end{bmatrix} \quad (\text{IV-117})$$

Case 2. Noncondensable gases are present. In this case, the mass-transfer rate of the noncondensable gas is zero by definition. However, the interfacial temperature in this case is unknown. Thus, the matrix equation is rewritten to solve for the N-1 unknown mass-transfer rates and the interfacial temperature. Thus, the equation is the following.

$$\begin{bmatrix} (1-x_{1i}) & -x_{1i} & -x_{1i} & \cdot & 0 \\ -x_{2i} & (1-x_{2i}) & -x_{2i} & \cdot & 0 \\ -x_{3i} & -x_{3i} & (1-x_{3i}) & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ h_{lg,1} & h_{lg,2} & h_{lg,3} & \cdot & (h_o A_i + h_{\infty} A_i) \end{bmatrix} \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ \cdot \\ T_i \end{bmatrix} = \begin{bmatrix} -g A_i (x_{1\infty} - x_{1i}) \\ -g A_i (x_{2\infty} - x_{2i}) \\ -g A_i (x_{3\infty} - x_{3i}) \\ \cdot \\ h_o A_i T_o + h_{\infty} A_i T_{\infty} \end{bmatrix}$$

(IV-118)

Step 4. After the matrix has been inverted and solved for the mass-transfer rates Γ_m , the rates are summed to yield a net mass flux $\Gamma = \sum_m \Gamma_m$. This net mass-transfer rate is used to calculate a correction factor, f_g , for the heat- and mass-transfer coefficients g^* and h^* . The corrected heat- and mass-transfer coefficients are

$$h_{\infty} = h^* f_g \quad \text{and} \quad g = g^* f_g ,$$

where

$$f_g = \frac{\frac{\Gamma}{(g^* A_i)}}{e^{\left[\frac{\Gamma}{(g^* A_i)} \right] - 1}} .$$

Step 5. If the mass-transfer rates Γ_m and the surface temperature T_i (if it is changing during the iteration) are converged to within a 1% change, then proceed to Step 7; if not, go to Step 6.

Step 6. In this step, the changing variables are updated.

Case 1. Noncondensable gases are not present. In this case, the interfacial temperature and mass concentrations remain fixed so that only the heat- and mass-transfer coefficients (h_{∞} and g) are updated. The calculation then proceeds directly to Step 3, Case 1.

Case 2. Noncondensable gases are present. The interfacial temperature is varying in this case so that a new updated temperature must be used. The T_i solved for in Eq. (III-223) could be used except that the new T_i often exceeds the boiling temperature. This is impossible if all N-1 materials exist in the condensed phase. To overcome this problem, a relaxation parameter α is introduced such that

$$T_i = \alpha T_i^{\text{old}} + (1 - \alpha) T_i^{\text{new}} \quad (\text{IV-119})$$

is the updated value of the interfacial temperature. The relaxation parameter currently depends on the dynamics of the phase-transition process. New interfacial mass fractions x_{mi} are calculated using Eqs. (III-223) and (III-224). The calculation procedure then returns to Step 3, Case 2.

Step 7. In this step, the assumptions made at the beginning of the iteration are checked for consistency. At the beginning of the iteration (Step 2), all species that exist in the vapor phase also are assumed to exist in the condensed phase with the exception of true noncondensable gases (such as air, noble gases, and so on). This need not be the case because a condensable species also can act as a noncondensable one. For example, consider simultaneous condensation of sodium, steel, and uranium dioxide vapor upon cold steel structure in the absence of fission gas. If the steel structure is sufficiently cold, all three species will condense. However, if the structure temperature were higher, sodium might not condense, and in fact, if a thin film of sodium were present on the structure, it would evaporate as the others condensed. The vaporization-condensation transition temperature cannot be predicted at the outset; hence, this step is required to verify that the vaporization rates (Γ_m positive) are consistent with that particular material existing in the condensed phase. If it is not consistent, then the inconsistent material must be treated as a noncondensable gas, and the calculations are repeated by returning to Step 2. If all the fluxes are consistent, the solution has converged.

The methods described in the above seven steps are for a set of given conditions including the free-stream temperature T_∞ , the condensed-phase temperature T_0 , the vapor-side heat-transfer coefficient h_∞ , the condensed-side heat-

transfer coefficient h_0 , and the mass fractions of the materials $x_{m,\infty}$. During a SIMMER-II calculation, these conditions can change rapidly (sometimes dramatically) in a fluid-dynamics time step. If such changes in conditions are not taken into account, errors can be introduced that can cause instabilities or unrealistic phenomena. To account for these changes, the SIMMER-II fluid-dynamics time step is divided into smaller time steps in which the free-stream variables are updated. During each of the smaller time steps, the mass-transfer rates and interfacial temperatures predicted by the above steady-state model are assumed to remain constant. The mass and energy equations for the mesh cell then are solved implicitly over the time step without accounting for any mass or energy convection. For a mesh cell having S surfaces and N mass-transfer rates at each surface, the vapor-field internal-energy equation is

$$\begin{aligned} \bar{\rho}_G \bar{c}_{vG} \frac{d}{dt} T_G = & \sum_{k=1}^S h_k A_k (T_{ki} - T_G) + \sum_{k=1}^S \sum_{m=1}^N \Gamma_{mk} c_{pGM} (T_{ki} - T_G) \\ & + \sum_{k=1}^S \sum_{m=1}^N \Gamma_{mk} R_M \alpha_G T_G \quad , \end{aligned} \quad (IV-120)$$

where

Γ_{mk} is the mass-transfer rate of material m on surface k ,

T_{ki} is the interfacial temperature of surface k , and

h_k is the vapor-side heat-transfer coefficient for surface k .

The vapor temperature in Eq. (IV-120) is evaluated at the end of the small time step. The resulting temperature is

$$T_G^{\kappa+1} = \frac{X}{\bar{Y}} \quad , \quad (IV-121)$$

where

$$X = \bar{\rho}_G \bar{c}_{vG} T_G + \delta t^\kappa \sum_{k=1}^S (h_k A_k T_{ki} + \sum_{m=1}^N \Gamma_{mk} c_{pGM} T_{ki}) \quad \text{and}$$

$$Y = \bar{p}_G \bar{c}_{vG} + \delta t^\kappa \sum_{k=1}^S [h_k A_k + \sum_{m=1}^N (\Gamma_{mk} c_{pGM} - \alpha_G \Gamma_{mk} R_M)]$$

and where $T_G^{\kappa+1}$ is the vapor temperature at the end of the small time step δt^κ .

Similar equations are obtained for the bulk temperatures of the liquid-field energy components except that there are no compressive heating terms. Thus,

$$T_{Lm}^{\kappa+1} = \frac{\bar{p}_{Lm} c_{vLM} T_{Lm}^\kappa + \delta t^\kappa (h_{Lm} A_{Lm} T_{Lmi} - c_{vLM} \sum_{k=1}^S \Gamma_{mk} T_{ki})}{\bar{p}_{Lm} c_{vLM} + \delta t^\kappa (h_{Lm} A_{Lm} - c_{vLM} \sum_{k=1}^S \Gamma_{mk})}, \quad (IV-122)$$

where

$T_{Lm}^{\kappa+1}$ is the bulk temperature of liquid-field energy component m at the end of the small time step,

$\bar{p}_{Lm}$ is the macroscopic density of liquid-field energy component m ,

c_{vLM} is the constant-volume specific heat of liquid-field energy component m ,

h_{Lm} is the liquid-side heat-transfer coefficient of liquid-field energy component m ,

A_{Lm} is the interfacial surface area per unit volume for liquid-field energy component m , and

T_{Lmi} is the interfacial temperature of liquid-field energy component m .

Finally, the energy equations for the solid surfaces have no mass source terms because sublimation is not considered, so

$$T_{Sm}^{\kappa+1} = \frac{\bar{p}_{Sm} c_{vSM} T_{Sm}^\kappa + \delta t^\kappa h_{Sm} A_{Sm} T_{Smi}}{\bar{p}_{Sm} c_{vSM} + \delta t^\kappa h_{Sm} A_{Sm}}, \quad (IV-123)$$

where

T_{Sm}^{k+1} is the bulk temperature of solid energy component m at the end of the small time step,

$\bar{\rho}_{Sm}$ is the macroscopic density of solid energy component m ,

c_{vSm} is the constant-volume specific heat of solid energy component m ,

h_{Sm} is the solid-side heat-transfer coefficient of solid energy component m ,

A_{Sm} is the interfacial surface area per unit volume of solid energy component m , and

T_{Smi} is the interfacial temperature of solid energy component m .

The small time step, δt^k , is chosen such that the advanced-time temperature for the vapor on the surfaces does not differ from the previous temperature by more than 10%. The advanced-time temperatures and densities then are used in the boundary layer treatment to compute the mass-transfer rates and interfacial temperatures for the next time step. This procedure is repeated until the SIMMER-II fluid-dynamics time step is completed. The average mass- and energy-transfer rates for the fluid-dynamics time step are computed from the difference between the end- and the beginning-of-time-step values of density and energy. These average rates then are passed through arrays to subroutine EXFL1 for solving the field equations.

9. Completion of the Exchange Functions. Following convergence of the phase-transition model, the momentum-exchange functions are evaluated. The fluid-structure exchange functions are evaluated as explained in Eqs. (III-141) to (III-165). The only required observations are as follows.

- (a) The velocity (liquid or vapor) used for a given cell (i,j) is obtained from the formula

$$|\vec{V}| = \frac{1}{2}[(v_{i,j+\frac{1}{2}} + v_{i,j-\frac{1}{2}})^2 + (u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j})^2]^{1/2}, \quad (\text{IV-124})$$

where V is an axial velocity component and U is a radial velocity component.

- (b) If the structure volume fraction exceeds an input quantity, ALDRG, the momentum-exchange functions are set to a large value (10^{12}). In other words, blockages are assumed to exist before extremely small volumes remain to accommodate fluid, which could cause numerical difficulties.

- (c) The maximum value allowed for these exchange functions is 10^{12} . Smaller values have been observed to allow unacceptable convection into blocked regions. Larger values can lead to convergence problems in the pressure iteration because of loss of significant figures.

The liquid-vapor momentum exchange function, K_{GL} , is evaluated from Eq. (III-168). Observations on its evaluation are as follows.

- (a) The vapor macroscopic density is the updated value obtained from the phase-transition model.
- (b) The liquid-vapor velocity difference is the minimum of either $|V_{GL}^n|$ from Eq. (IV-65) or the $|V_{GL}^{n+1}|$ obtained from Eq. (IV-54). The general idea is to avoid oscillations resulting from use of an explicitly calculated velocity difference.
- (c) Besides a maximum of 10^{10} , a minimum value of 10^{-4} has been placed on the vapor-liquid momentum exchange function. This value is used to avoid difficulties with the pressure iteration if only small quantities of liquid exist in a computational cell.

When the momentum-exchange functions are known, the drag heating terms can be evaluated from Eqs. (III-169) to (III-174). Observations on these calculations are as follows.

- (a) The calculation is performed only for two-phase cells that have entered the vaporization-condensation model. In addition, a nonnegligible liquid volume fraction must exist.
- (b) The minimum of K_{GL}^n (this time step) and K_{GL}^{n-1} (previous time step) is used to avoid excessive heating resulting from an increase in K_{GL} without an appropriate adjustment in the relative velocities. The relative liquid-vapor velocity used is that previously obtained for the K_{GL}^n calculation.
- (c) The heat-transfer coefficients come from the vaporization-condensation model. The vapor-side heat transfer coefficients are those without rate corrections, such as those given by Eq. (III-139).

After evaluation of the drag heating terms, $\tilde{e}_{Lm}^{n+1}$ and $\tilde{e}_{Sm}^{n+1}$ are updated to reflect the additional heat transfer. At this point in the calculation, the can-wall set-over model again is accommodated. If ICW is 1, the extra heat transfer to the can-wall must be multiplied by $V_{ij}/V_{i-1,j}$. If ICW is 2, the extra heat is multiplied by $V_{ij}/V_{i+1,j}$ before adding it to $\tilde{q}_{H,S4,i+1,j}$. If ICW is 3, the extra energy simply is ratioed by density (based on $\rho_{s6,i-1,j}$ and

$\bar{p}_{s6,i+1,j}$) toward the appropriate directions and then multiplied by the volumetric weighting terms.

Last, the melting-freezing phase transition rates and the fission gas release rates are calculated as explained in Sec. III.D.6.

D. Plug Modeling

After the exchange functions have been calculated, the structural changes caused by plug motion are evaluated. A summary of the plug model is in Chap. III.D.5.f. The FORTRAN is activated by the *IF DEF, PLUG option. As stated in Chap. III, plug motion starts after an input breakaway force is exceeded.

The plug of interest is assumed to be located axisymmetrically and to have a known radius and mass. A node structure is applied to the problem from which the location of the bottom interface of the plug is described in terms of the radial node number of the outermost radial node and the axial node number of the bottom of the plug. This arrangement is shown in Fig. 12. A plug simulation region generally is defined from the plug lower interface to the top of the (RCB).

A special mesh point set and parameter region are assigned to this simulation region to accommodate plug interface tracking and fluid dynamic coupling. For a standard LMFBR problem, the region should be set up to contain 5% by volume liquid sodium and 95% "no-flow" volume. It also should be single-phase liquid to establish the infinite drag characteristics in the undisturbed part of the region (ALDRG should be set to 0.949).

The lower interface position is calculated from a reference location (ZPLUG = 0.0) at the bottom of cell row JSO as shown in Fig. 13. Because the Δz_j of the cells in the plug simulation region may vary, it is necessary to establish current values for the position of the top (ZTOP) and bottom (ZBOT) of the cell in which the plug interface exists (JPLUGI). As the interface passes either upward or downward into another cell, JPLUGI, ZBOT, and ZTOP are updated.

The plug dynamics are computed in an explicit manner within SIMMER-II. The coupling to the fluid dynamics is accomplished by a time-dependent structure volume fraction (time-dependent no-flow volume). As the plug dynamics model predicts motion of the lower plug interface, "effective" structure is removed to simulate the additional volume available to the impinging fluid. The SIMMER-II fluid dynamics then follow a natural course in which the fluid attempts to maintain contact with the moving plug to produce an accelerating force.

Fig. 12.
Node structure for defining the initial plug position.

Fig. 13.
Plug interface tracking scheme.

If the fluid cannot maintain contact, local cavitation occurs, and a downward force on the plug results. The actual addition of the no-flow volume is in Sec. IV.E, which will be described after Eq. (IV-14f).

Because the coupling of plug and fluid dynamics is explicit, numerical problems tend to develop when the plug interface passes into a new cell, especially when the plug velocity is high. The liquid volume fraction in the new cell is on the order of 5%, and the cell contains single-phase liquid. Therefore, a substantial pressure fluctuation can occur in this cell as structure rapidly is removed (plug interface motion), and liquid from the cell below enters by convection. To help stabilize this problem, the plug motion (after completion of the exchange functions) is updated only to the beginning of the current time step. A time step control then is applied to prevent the structure volume change from being larger than a fraction of the liquid volume fraction in the next time step. Then when the liquid volume fraction is small, the time step will be smaller. To avoid the influences of these numerical fluctuations on the plug dynamics, the pressure assumed to act on the plug is taken from the cell below the interface until the interface passes through 20% of the JPLUGI cell.

Changes to the SIMMER-II implicit fluid dynamics algorithm in Sec. IV.F are required to prevent nonphysical cavitation. The problem arises from the formulation of the momentum equations. The area over which the pressure gradient acts is the minimum area for the two half-cells that form the momentum cell. This is shown in Fig. 14. SIMMER-II would use A_3 as this area. As a result, the force driving fluid into cell JPLUGI is continually too small. The physical situations should be as shown on the left side of Fig. 14 in which A_2 equals A_1 . The changes made in SIMMER-II provide for the maximum area from either cell JPLUGI or JPLUGI-1 to be used. This is discussed after Eq. (IV-188).

Because the SIMMER-II geometric model for this type of problem includes the reactor head and some portion of the RCB, the reactor cover gas and pool impact plane are no longer at the upper boundary of the calculative mesh. Thus, editing changes are required. The user must input the mesh boundaries (radial and axial) of the reactor cover gas region. The impact plane then is assumed to be the upper boundary of this region. The cover gas volume is calculated by integrating the vapor volume within this defined region. The impulse to the reactor head is obtained by integrating the pressure at the impact plane in space and time. The instantaneous average pressure acting on the head also is assessed at this impact plane.

Fig. 14.
SIMMER-II treatment of the fluid dynamics at the plug interface.

When the ejected plug is displaced by a distance equal to the thickness of the reactor head, material ejection into the RCB can begin. Edits are provided that give the cumulative amount of sodium (liquid plus vapor), fuel liquid, fuel vapor, fuel particles, and stainless steel (vapor plus liquid plus particles) ejected. These time integrals are performed at the top boundary of the uppermost node in the reactor head (input by the user as JSOT). The totals are output at each time step as part of the short prints.

Some additional force edits also have been included. These edits give the instantaneous total upward and downward forces acting on the vessel supports. The up-force is obtained by integrating the instantaneous pressure distribution over the reactor head impact plane. The down-force is obtained by integrating the instantaneous pressure distribution at the lower surface of the active core and at the upper surface of the annular shield region (from the active core radial boundary to the vessel wall). However, the integration surfaces are input-controlled to provide user flexibility in this regard. These forces also are output at each time step as part of the short prints. Pressures at the axis of the reactor and the vessel wall at the impact plane also are output each time step. The format for the variables printed is given in Sec. VI.E. The input description for the plug model is included in Appendix Q.

E. Explicit Calculation

In the explicit phase of the calculation, the mass-continuity and energy equations are solved explicitly for all field components. For the structure and the vapor fields in single-phase cells, the resulting macroscopic densities represent the final advanced-time solution. The remaining densities are updated as part of the implicit calculation. Except for the vapor, the energies calculated in the explicit phase are the final advanced-time values. Furthermore, portions of the liquid- and vapor-field momentum equations are computed explicitly.

1. Explicit Continuity Equations. The advanced-time macroscopic densities of the structure-field density components are computed as

$$\bar{\rho}_{sm}^{n+1} = \bar{\rho}_{sm}^n - \Delta t \Gamma_{sm} \quad , \quad m = 1, \dots, \text{NCSR} \quad . \quad (\text{IV-125})$$

No convection terms are present in this equation because the structure field is stationary.

The liquid-field density component continuity equations are solved according to

$$\bar{\rho}_{\ell m} = \bar{\rho}_{\ell m}^n - \Delta t \left[\Gamma_{\ell m} + \frac{\Delta_i \langle \bar{\rho}_{\ell m} r^{\xi} U_L \rangle_j^n}{r_i^{\xi} \Delta r_i} + \frac{\Delta_j \langle \bar{\rho}_{\ell m} v_L \rangle_i^n}{\Delta z_j} \right] ,$$

$$m = 1, \dots, NCLR , \quad (IV-126)$$

where Δ_i and Δ_j are shorthand notations for spatial differencing.

For a space-dependent function f , these differencing operators are applied in the following manner,

$$\Delta_i f = f_{i+\frac{1}{2},j} - f_{i-\frac{1}{2},j} , \quad (IV-127)$$

and

$$\Delta_j f = f_{i,j+\frac{1}{2}} - f_{i,j-\frac{1}{2}} . \quad (IV-128)$$

The brackets ($\langle \rangle$) in Eq. (IV-126) are used to denote donor-cell differencing of the convective fluxes. A blending of partial with full donor-cell differencing is defined to mitigate truncation errors and to maintain solution stability. For the space-dependent function f , the donor-cell difference scheme for the convective flux $\langle f r^{\xi} U \rangle$ on the $i+1/2, j$ interface between mesh cell i, j and mesh cell $i+1, j$ is

$$\langle f r^{\xi} U \rangle_{i+\frac{1}{2}} = r_{i+\frac{1}{2}}^{\xi} U_{i+\frac{1}{2},j} \left[\left(\frac{\frac{1}{2} \Delta r_{i+1}}{\Delta r_{i+\frac{1}{2}}} + \eta_{i+\frac{1}{2}} \right) f_{ij} + \left(\frac{\frac{1}{2} \Delta r_i}{\Delta r_{i+\frac{1}{2}}} - \eta_{i+\frac{1}{2}} \right) f_{i+1,j} \right] ,$$

$$(IV-129)$$

where

$$\eta_{i+\frac{1}{2}} = \frac{a_0}{\Delta r_{i+\frac{1}{2}}} \left\{ \begin{array}{ll} \Delta r_i & , \quad U_{i+\frac{1}{2},j} \geq 0 \\ -\Delta r_{i+1} & , \quad U_{i+\frac{1}{2},j} < 0 \end{array} \right\} + \frac{b_0 \Delta t}{\Delta r_{i+\frac{1}{2}}} U_{i+\frac{1}{2},j} \quad , \quad (IV-130)$$

where a_0 and b_0 are input coefficients used to obtain the desired amounts of full and partial donor-cell differencing that we require to satisfy $a_0 + b_0 = 1/2$. Generally, $a_0 = 1/2$ and $b_0 = 0$ are used in SIMMER-II to define full donor cell differencing. This choice yields a positive definite result, that is, $\tilde{\rho}_{\ell m} \geq 0$, if the effect of $\Gamma_{\ell m}$ is neglected. Partial donor-cell differencing can be obtained with $a_0 = 0$ and $b_0 = 1/2$, which linearly interpolates the f function as donor-cell convected. When $b_0 \neq 0$, the above general form of Eqs. (IV-129) and (IV-130) is evaluated. However, when $b_0 = 0$, the following more efficient full donor cell form,

$$\langle f \rangle_{i+\frac{1}{2}} = r_{i+\frac{1}{2}}^{\xi} U_{i+\frac{1}{2},j} \left\{ \begin{array}{ll} f_{ij} & , \quad U_{i+\frac{1}{2},j} \geq 0 \\ f_{i+1,j} & , \quad U_{i+\frac{1}{2},j} < 0 \end{array} \right\}, \quad (IV-131)$$

is evaluated instead. The convection terms in other conservation equations that follow also are programmed in both forms to use the more efficient full donor cell form when chosen by the user. Further, full donor cell differencing as assumed in the evaluation of convection outflow for the DTL(8) time-step control (Sec. IV.G.5.h) and in the evaluation of $\Gamma_{g6,extra}$, as discussed after Eq. (IV-135).

Note that the macroscopic density on the left side of Eq. (IV-126) does not have an $n+1$ superscript. These densities are used for updating the liquid field internal energy only. The final advanced-time macroscopic density of each liquid field component is determined by a later implicit calculation.

The vapor continuity equations are updated for single-phase cells as discussed in Sec. IV.E.4. The solution for two-phase cells is discussed in Sec. IV.E.5.

The most important tasks in solving Eqs. (IV-125) and (IV-126) and the vapor continuity equations is obtaining the source terms (the Γ_{sm} , $\Gamma_{\ell m}$, and the Γ_{gm}). Three tasks are required.

- (a) The combined mass-transfer rates for the energy components calculated by the exchange functions are separated into the density component mass-transfer rates.
- (b) Any adjustments in these rates as a result of crust or pellet breakup are performed.
- (c) Mass sources are defined at two-phase/single-phase boundaries to eliminate mass conservation errors made in the previous implicit calculation.

The combined mass-transfer rates between the various energy components are passed by the array variables as shown in Table XII. The first step in making

TABLE XII
STORAGE VARIABLES FOR MASS-TRANSFER RATES

<u>FORTTRAN Variable</u>	<u>Mass-Transfer Rate</u>	
	<u>Posi tive FORTTRAN Variable</u>	<u>Negative FORTTRAN Variable</u>
FS(IJ)	Γ_{S1L1}	
FS(IJ+1)	Γ_{S2L1}	Γ_{L1S2}
FS(IJ+2)	Γ_{S3L2}	Γ_{L2S3}
FS(IJ+3)	Γ_{S4L2}	Γ_{L2S4}
FS(IJ+4)	Γ_{S5L4}	Γ_{L4S5}
FS(IJ+5)	Γ_{s8s9}	
FS(IJ+6)	Γ_{s9g6}	
FL(IJ)	Γ_{L5L1}	Γ_{L1L5}
FL(IJ+1)	Γ_{L6L2}	Γ_{L2L6}
GAMCE(IJ)	Γ_{G1L1}	Γ_{L1G1}
GAMCE(IJ+1)	Γ_{G2L2}	Γ_{L2G2}
GAMCE(IJ+2)	Γ_{G3L3}	Γ_{L3G3}
GAMCE(IJ+3)	Γ_{G4L4}	Γ_{L4G4}

the required separation is to determine the mass fractions of fertile and fissile fuel for each of the fuel density components. These are

$$x_{smSk} = \frac{\bar{\rho}_{sm}}{\bar{\rho}_{Sk}}, \quad \begin{array}{ll} m = 1, 2 & k = 1, \\ m = 3, 4 & k = 2, \end{array} \quad (\text{IV-132})$$

$$x_{lmLk} = \frac{\bar{\rho}_{lm}}{\bar{\rho}_{Lk}}, \quad \begin{array}{ll} m = 1, 2 & k = 1, \text{ and} \\ m = 6, 7 & k = 5, \end{array} \quad (\text{IV-133})$$

$$x_{gmG1} = \frac{\bar{\rho}_{gm}}{\bar{\rho}_{G1}}, \quad m = 1, 2, \quad (\text{IV-134})$$

where

x_{xmSk} is the mass fraction of fertile fuel ($m = 1$ or $m = 3$) or fissile fuel ($m = 2$ or $m = 4$) in fabricated fuel ($k = 1$) or frozen fuel ($k = 2$),

x_{lmLk} is the mass fraction of fertile fuel ($m = 1$ or $m = 6$) or fissile fuel ($m = 2$ or $m = 7$) in the liquid fuel ($k = 1$) or solid fuel particles ($k = 5$), and

x_{gmG1} is the mass fraction of fertile fuel ($m = 1$) or fissile fuel ($m = 2$) in fuel vapor.

These mass fractions are used to separate the combined energy component mass-transfer rates into the individual mass-transfer rates shown in Tables XIII through XV. The individual mass-transfer rates are recombined to form the density component total mass-transfer rates shown in Tables IV, V, and XV. These total rates are those used in task (a). In this context, we should note that the FS arrays only contain the fraction of failed structure that is pure liquid. Multiplication of these quantities by $(1 - x_{Fm})/x_{Fm}$, where x_{Fm} is an input fraction, is required to get the particle contributions indicated in Table XIII. If sufficient material does not exist for these particle transfer rates, the energy

TABLE XIII
DEFINITION OF THE STRUCTURE-FIELD DENSITY COMPONENT
INDIVIDUAL MASS-TRANSFER RATES

$\Gamma_{s1\ell1}$	$=$	$x_{s1S1} \Gamma_{S1L1}$
$\Gamma_{s1\ell6}$	$=$	$(1 - x_{F1}) x_{s1S1} \Gamma_{S1L1}/x_{F1}$
$\Gamma_{s2\ell2}$	$=$	$x_{s2S1} \Gamma_{S1L1}$
$\Gamma_{s2\ell7}$	$=$	$(1 - x_{F1}) x_{s2S1} \Gamma_{S1L1}/x_{F1}$
$\Gamma_{s3\ell1}$	$=$	$x_{s3S2} \Gamma_{S2L1}$
$\Gamma_{s3\ell6}$	$=$	$(1 - x_{F2}) x_{s3S2} \Gamma_{S2L1}/x_{F2}$
$\Gamma_{s4\ell2}$	$=$	$x_{s4S2} \Gamma_{S2L1}$
$\Gamma_{s4\ell7}$	$=$	$(1 - x_{F2}) x_{s4S2} \Gamma_{S2L1}/x_{F2}$
$\Gamma_{s5\ell3}$	$=$	Γ_{S3L2}
$\Gamma_{s5\ell8}$	$=$	$(1 - x_{F3}) \Gamma_{S3L2}/x_{F3}$
$\Gamma_{s6\ell3}$	$=$	Γ_{S4L2}
$\Gamma_{s6\ell8}$	$=$	$(1 - x_{F4}) \Gamma_{S4L2}/x_{F4}$
$\Gamma_{s7\ell5}$	$=$	Γ_{S5L4}

transfer arrays have been set up to anticipate that material not transferred as liquid will be transferred as particles. Finally, we must note that the exchange rates were calculated using an estimated time-step size, $\Delta\tilde{t}$. The time-step size used in the explicit calculations, Δt , may be shorter. Consequently, the maximum rate for disappearance of any structural material is $\bar{p}_{vm}/\Delta\tilde{t}$.

For task (b), fabricated fuel and crust breakup possibilities discussed in Sec. III.D.6.a are added to the Table XIII rates. It should be noted that for fabricated fuel breakup, a $\Gamma_{s1\ell6}$ and $\Gamma_{s2\ell7}$ addition, any associated intergranular fission gas is released to the vapor field. The intragranular fission gas is ignored, except that the current Γ_{s8s9} contribution is turned into a Γ_{s9g6} contribution. It also should be noted that crust breakup occurs when

TABLE XIV
DEFINITION OF THE LIQUID-FIELD DENSITY COMPONENT
INDIVIDUAL MASS-TRANSFER RATES

$\Gamma_{1s3} =$	$x_{1L1} \Gamma_{L1S2}$
$\Gamma_{116} =$	$x_{1L1} \Gamma_{L1L5}$
$\Gamma_{1g1} =$	$x_{1L1} \Gamma_{L1G1}$
$\Gamma_{2s4} =$	$x_{2L1} \Gamma_{L1S2}$
$\Gamma_{217} =$	$x_{2L1} \Gamma_{L1L5}$
$\Gamma_{2g2} =$	$x_{2L1} \Gamma_{L1G1}$
$\Gamma_{3s5} =$	Γ_{L2S3}
$\Gamma_{3s6} =$	Γ_{L2S4}
$\Gamma_{318} =$	Γ_{L2L6}
$\Gamma_{3g3} =$	Γ_{L2G2}
$\Gamma_{4g4} =$	Γ_{L3G3}
$\Gamma_{5s7} =$	Γ_{L4S5}
$\Gamma_{5g5} =$	Γ_{L4G4}
$\Gamma_{611} =$	$x_{6L5} \Gamma_{L5L1}$
$\Gamma_{712} =$	$x_{7L5} \Gamma_{L5L1}$
$\Gamma_{813} =$	Γ_{L6L2}

$$\frac{\bar{\rho}_{s6}^n}{\Delta t} x_{F4} - 0.1 \frac{\text{kg}}{\text{m}^3 \cdot \text{s}} < \Gamma_{S4L2}$$

but while $\Gamma_{S4L2} > 0$, in other words, when the last mass of a can-wall is predicted to melt.

For task (c), we note that the current SIMMER-II algorithm maintains a constant vapor density in single-phase cells during the implicit update of vapor densities. Any vapor convected from a two-phase cell into a single-phase cell is lost; any vapor convected from a single-phase cell into a two-phase cell results in a mass gain. To conserve mass, extra sources or sinks are defined on

TABLE XV
DEFINITION OF THE VAPOR-FIELD DENSITY COMPONENT
INDIVIDUAL AND TOTAL MASS-TRANSFER RATES

$$\begin{aligned}\Gamma_{g1} &= \Gamma_{g1\ell1} - \Gamma_{\ell1g1} = x_{g1G1} \Gamma_{G1L1} - x_{\ell1L1} \Gamma_{L1G1} \\ \Gamma_{g2} &= \Gamma_{g2\ell2} - \Gamma_{\ell2g2} = x_{g2G1} \Gamma_{G1L1} - x_{\ell2L1} \Gamma_{L1G1} \\ \Gamma_{g3} &= \Gamma_{g3\ell3} - \Gamma_{\ell3g3} = \Gamma_{G2L2} - \Gamma_{L2G2} \\ \Gamma_{g4} &= \Gamma_{g4\ell4} - \Gamma_{\ell4g4} = \Gamma_{G3L3} - \Gamma_{L3G3} \\ \Gamma_{g5} &= \Gamma_{g5\ell5} - \Gamma_{\ell5g5} = \Gamma_{G4L4} - \Gamma_{L4G4} \\ \Gamma_{g6} &= -\Gamma_{s9g6}\end{aligned}$$

the subsequent time step. For condensable vapor, these sources are additions to $\Gamma_{\ell m}$ in single-phase cells, assuming any vapor entering is condensed and any vapor exiting is replaced by vaporizing liquid. The $\Gamma_{\ell m, \text{extra}}$ contributions are

$$\begin{aligned}\Gamma_{\ell m, \text{extra}} &= \frac{\Delta t^{n-1}}{\Delta t^n} \left[\frac{\xi_{i-1,j} \langle \bar{\rho}_{gm} r^{\xi} U_G \rangle_{i-\frac{1}{2},j}^n - \xi_{i+1,j} \langle \bar{\rho}_{gm} r^{\xi} U_G \rangle_{i+\frac{1}{2},j}^n}{r_i^{\xi} \Delta r_i} \right. \\ &\quad \left. + \frac{\xi_{i,j-1} \langle \bar{\rho}_{gm} V_G \rangle_{i,j-\frac{1}{2}}^n - \xi_{i,j+1} \langle \bar{\rho}_{gm} V_G \rangle_{i,j+\frac{1}{2}}^n}{\Delta z_j} \right], \quad (\text{IV-135})\end{aligned}$$

where

Δt^{n-1} is the previous cycle time-step,

ξ is 1 if the identified neighboring cell is two-phase, and

ξ is 0 if the identified neighboring cell is single-phase.

For noncondensable vapor (fission gas), the source appears in the neighboring two-phase cell representing vapor convected into the single-phase cell in the previous time step. Single-phase cells do not contain fission gas; consequently, no gas can be convected out. The $\Gamma_{g6, \text{extra}}$ source otherwise is com-

puted similarly to Eq. (IV-135), but pure donor cell differencing is assumed. Implementation of interpolated differencing has not been performed because of the nonphysical convection of fission gas out of a single-phase cell possible with this option.

2. Explicit Energy Equations. The next step of the explicit calculation involves determining the energies for the liquid- and the structure-field energy components. The advanced-time energy density for each liquid-field energy component is given by

$$\begin{aligned}
 (\bar{\rho}_{Lm} e_{Lm})^{n+1} = & (\bar{\rho}_{Lm} e_{Lm})^n - \Delta t \left(\frac{\Delta_i \langle \bar{\rho}_{Lm} e_{Lm} r_i^\xi U_L \rangle_j^n}{r_i^\xi \Delta r_i} + \frac{\Delta_j \langle \bar{\rho}_{Lm} e_{Lm} v_L \rangle_i^n}{\Delta z_j} \right) \\
 & + \Delta t (Q_{Lm} + N Q_{NLm}) + \Delta t \bar{\rho}_{Lm}^n v_L \left\{ 2 \left[\left(\frac{\Delta_i U_{Lj}^n}{\Delta r_i} \right)^2 + \xi \left(\frac{U_{Li}^n}{r_i^\xi} \right)^2 \right. \right. \\
 & \left. \left. + \left(\frac{\Delta_j v_{Li}^n}{\Delta z_j} \right)^2 \right] - \frac{2}{3} \left[\left(\frac{\Delta_i (r_i^\xi U_L^n)_j}{r_i^\xi \Delta r_i} \right) + \frac{\Delta_j v_{Li}^n}{\Delta z_j} \right]^2 \right. \\
 & \left. + \left(\frac{\Delta_j U_{Li}^n}{\Delta z_j} + \frac{\Delta_i v_{Lj}^n}{\Delta r_i} \right)^2 \right\} \quad m = 1, \dots, NCLE \quad (IV-136)
 \end{aligned}$$

Again, the spatial-difference terms represent the convection of the liquid-field energy component energy across the mesh cell interfaces. The term Q_{Lm} contains all the energy from the exchange functions calculated previously. This includes energy because of mass changes from fabricated fuel and crust breakup and the energy associated with the sources because of the single-phase/two-phase interface source adjustments. The particles from fabricated fuel and crust breakup enter the liquid fuel at the energy they possessed as solids. The condensable interfacial liquid sources are given the vapor energy, that is, $\bar{\rho}_{gm}$ is replaced by $\bar{\rho}_{Gm} e_G$ in the energy equivalent to Eq. (IV-135). Extra fission gas sources in two-phase interfacial cells are discussed in conjunction with the vapor field equations. The term NQ_{NLm} represents the nuclear heating rate to the liquid-field energy component. The remaining terms define the optional viscous

heating rate and are evaluated only for the first four liquid energy components. The differencing of the viscous heating rate is straightforward because of the location of velocities at cell boundaries. The exception is the final term. It is differenced according to

$$\frac{\Delta_j U_{Li}^n}{\Delta z_j} = \frac{U_{L,i+\frac{1}{2},j+1} + U_{L,i-\frac{1}{2},j+1} - U_{L,i+\frac{1}{2},j-1} - U_{L,i-\frac{1}{2},j-1}}{\Delta z_{j-1} + 2\Delta z_j + \Delta z_{j+1}}, \quad (\text{IV-137})$$

and

$$\frac{\Delta_i V_{Lj}^n}{\Delta r_i} = \frac{V_{L,i+1,j+\frac{1}{2}} + V_{L,i+1,j-\frac{1}{2}} - V_{L,i-1,j+\frac{1}{2}} - V_{L,i-1,j-\frac{1}{2}}}{\Delta r_{i-1} + 2\Delta r_i + \Delta r_{i+1}}. \quad (\text{IV-138})$$

The advanced-time specific internal energy for the liquid-field energy component is

$$e_{Lm}^{n+1} = \frac{(\bar{\rho}_{Lm} e_{Lm})^{n+1}}{\bar{\rho}_{Lm}}, \quad m = 1, \dots, \text{NCLE}, \quad (\text{IV-139})$$

where the macroscopic density $\bar{\rho}_{Lm}$ of the liquid-field energy component is obtained from the macroscopic densities computed from Eq. (IV-126).

In a similar procedure for the structure-field density components, the energy density is given as

$$(\bar{\rho}_{Sm} e_{Sm})^{n+1} = (\bar{\rho}_{Sm} e_{Sm})^n + \Delta t (Q_{Sm} + N Q_{NSm}), \quad m = 1, \dots, \text{NCSE}. \quad (\text{IV-140})$$

The Q_{Sm} term is from the exchange processes, and the final term represents the nuclear heating rate to the structure-field energy component. The specific internal energy is computed to be

$$e_{Sm}^{n+1} = \frac{(\bar{\rho}_{Sm} e_{Sm})^{n+1}}{\bar{\rho}_{Sm}^{n+1}}, \quad m = 1, \dots, \text{NCSE}, \quad (\text{IV-141})$$

where $\bar{\rho}_{Sm}^{n+1}$ is the advanced-time macroscopic density of structure-field energy component m evaluated by Eq. (IV-125).

3. Advanced-Time Volume Fractions. The liquid-field and structure-field energy component energies determined in Eqs. (IV-139) and (IV-141) are used to calculate the melt fractions for the energy components. For the liquid-field solid components, the melt fraction is computed as

$$f_{Lm}^{n+1} = \begin{cases} 1 & , \text{ if } e_{Lm}^{n+1} \geq e_{\text{Liq},M} \\ \frac{e_{Lm}^{n+1} - e_{\text{Sol},M}}{h_{f,M}} & , \text{ if } e_{\text{Liq},M} > e_{Lm}^{n+1} > e_{\text{Sol},M} \\ 0 & , \text{ if } e_{Lm}^{n+1} \leq e_{\text{Sol},M} \end{cases} \quad m = 5, 6, \quad (\text{IV-142})$$

where

$e_{\text{Sol},M}$ is the solidus energy of material M assigned to liquid-field energy component m and

$h_{f,M}$ is the heat of fusion of material M .

The effective microscopic densities ρ_{Lm}^{n+1} of the solid density components in the liquid field are calculated as

$$\frac{1}{\rho_{Lm}^{n+1}} = \frac{f_{Lm}^{n+1}}{\rho_{Lm}} - \frac{1 - f_{Lm}^{n+1}}{\rho_{Lsm}}, \quad m = 6, 7, \text{NCLR} = 8, \quad (\text{IV-143})$$

where

$f_{\ell m}^{n+1}$ is the melt fraction of liquid-field density component m ($f_{\ell 6}^{n+1} = f_{L5}$, $f_{\ell 7}^{n+1} = f_{L5}$, and $f_{\ell 8}^{n+1} = f_{L6}$),

$\rho_{\ell \ell m}$ is the input microscopic density of the pure liquid phase of liquid-field density component m , and

$\rho_{\ell sm}$ is the input microscopic density of the pure solid phase of liquid-field density component m .

The microscopic densities $\rho_{\ell m}^{n+1}$ of the liquid density components in the liquid field are $\rho_{\ell \ell m}$ for liquid control material ($m = 5$). For liquid fuel, steel, and sodium, updated component liquid temperatures are first calculated from the e_{Lm}^{n+1} using the inverse of Eq. (III-43) or using the algorithm described in Eq. (IV-42) to Eq. (IV-45). The T_{Lm}^{n+1} then are used to get $\rho_{\ell m}^{n+1}$ by Eq. (III-71).

The effective microscopic densities and the estimated macroscopic densities of the liquid-field density components from Eq. (IV-126) then are used to compute the advanced-time liquid-field volume fraction:

$$\alpha_L^{n+1} = \sum_{m=1}^{NCLR} \tilde{\rho}_{\ell m} (1 - \alpha_o) / \rho_{\ell m}^{n+1} . \quad (IV-144)$$

Because of potential difficulties in the SIMMER-II formulation when very small amounts of liquid are present,* two tests are performed at this stage in the numerical algorithm. First, α_L^{n+1} is compared with an input constant, ALPMIN, which is typically 10^{-4} . If α_L^{n+1} is less than ALPMIN, the liquid-vapor exchange function, K_{GL} , is set to 10^{10} to lock the two momentum fields together. If α_L^{n+1} is less than 1% of ALPMIN, steel particles are introduced to increase α_L^{n+1} to $0.01 \cdot \text{ALPMIN}$. After determination of the liquid-field volume fractions, the liquid-field compressibility is estimated by the expression

$$\frac{1}{\tilde{G}_{Lij}} = \sum_{m=1}^{NCLR} \frac{\tilde{\rho}_{\ell m}}{(\rho_{\ell m}^{n+1} c_{\ell m}^2)} , \quad (IV-145)$$

*Aside from the difficulty of using neutron diffusion theory in a near vacuum, the explicit evaluation of the convective terms in the momentum equation can lead to problems if the liquid mass is small. Momentum can be convected without associated mass transport, and reduction in liquid density by an order of magnitude or more (for example, because of vaporization) can lead to large liquid velocity increases.

where

$\tilde{G}_{Lij}^{-1} = \left(\frac{\partial \bar{p}}{\partial p} \right)$ is the estimated liquid-field compressibility in mesh cell (i,j) and

$c_{\ell m}^2$ is the square of the sonic velocity for liquid-field density component m.

The structure-field volume fraction is obtained using a similar procedure. First, the melt fractions of the structure-field energy components are calculated as

$$f_{sm}^{n+1} = \begin{cases} 1 & , \text{ if } e_{sm}^{n+1} \geq e_{Liq,M} \\ \frac{e_{sm}^{n+1} - e_{Sol,M}}{h_{f,M}} & , \text{ if } e_{Liq,M} > e_{sm}^{n+1} > e_{Sol,M} \quad m = 1, 2, \dots, NCSE = 5 \\ 0 & , \text{ if } e_{sm}^{n+1} \leq e_{Sol,M} \end{cases} \quad (IV-146)$$

Then the effective microscopic density of each structure field density component is determined to be

$$\frac{1}{\rho_{sm}^{n+1}} = \frac{f_{sm}^{n+1}}{\rho_{s\ell m}} + \frac{1 - f_{sm}^{n+1}}{\rho_{ssm}} \quad , \quad m = 1, \dots, 7 \quad , \quad (IV-147)$$

where

f_{sm}^{n+1} is the melt fraction of structure-field density component m
 $(f_{s1}^{n+1} = f_{S1}^{n+1}, f_{s2}^{n+1} = f_{S1}^{n+1}, f_{s3}^{n+1} = f_{S2}^{n+1}, f_{s4}^{n+1} = f_{S2}^{n+1}, f_{s5}^{n+1} = f_{S3}^{n+1},$
 $f_{s6}^{n+1} = f_{S4}^{n+1}, f_{s7}^{n+1} = f_{S5}^{n+1})$,

$\rho_{s\ell m}$ is the input microscopic density of the pure liquid phase of structure-field density component m, and

ρ_{ssm} is the input microscopic density of the pure solid phase of structure-field density component m.

The microscopic density ρ_{sm}^{n+1} of the two fission gas components in the structure-

field is ρ_{ssm} for $m = 8, 9$. Finally, the structure-field volume fraction is computed as

$$\tilde{\alpha}_S^{n+1} = \frac{\text{NCSR-2 } \bar{p}_{sm}^{n+1}}{\sum_{m=1} \rho_{sm}^{n+1}} . \quad (\text{IV-148})$$

The two fission gas components currently are not included in the summation. The tilde value for the structure-field volume fraction is necessary because the nonflow volume fractions are yet to be included. As shown in Eqs. (III-7) and (III-8), the cladding nonflow contribution is $\alpha_{nf} \min[\bar{p}_{s5}, (\bar{p}_{s5})_{fail}] / (\bar{p}_{s5})_{fail}$. When the PLUG option is defined, α_{nf} is mesh-cell as well as parameter-region dependent in the region where a plug exists. When the plug motion can define α_{nf} , the $(\bar{p}_{s5})_{fail}$ input parameter generally is set to a small value to avoid distortion in having to define a large $\bar{p}_{s5}$. The can-wall nonflow contribution, $\alpha_{nf,cw}$, is added whenever radial velocities are not allowed, in other words, when FAIL3 is greater than one-half. (See Table VII.) Addition of these two contributions when appropriate produces α_S^{n+1} .

At this point, the advanced-time vapor volume fraction is obtained as

$$\alpha_G^{n+1} = 1 - \alpha_S^{n+1} - \alpha_L^{n+1} . \quad (\text{IV-149})$$

Because this α_G multiplies the pressure gradient in the vapor momentum equation, it must be positive. Therefore, the minimum value of α_G^{n+1} allowed is 10^{-6} . However, the real α_G is required to compute any initial single-phase pressure estimate. For this purpose, the actual α_G^{n+1} from Eq. (IV-149) is saved in a temporary location as $\tilde{\alpha}_G^{n+1}$. A second vapor volume fraction, α_G^{nf} , also is computed based on the final liquid densities from the pressure iteration of the previous time step and the interfield mass transport estimated to take place in the current time step. The three quantities α_G^{n+1} , α_G^{nf} , and α_G^n are used to determine whether a cell should be treated using single- or two-phase flow assumptions for the remainder of the calculation. In general, the flow is treated as two-phase if $\alpha_G^{n+1} \geq \alpha_0 (1 - \alpha_S^{n+1})$ and either $\alpha_G^{nf} \geq \alpha_0 (1 - \alpha_S^{n+1})$ or $\alpha_G^n \geq \alpha_0 (1 - \alpha_S^n)$. Otherwise, the flow is single-phase. However, if the details are examined, eight cases exist. These are summarized in Table XVI.

TABLE XVI

DECISION MATRIX FOR THE TWO-PHASE/SINGLE-PHASE TRANSITION IN EXFLUD

Number	$\alpha_G^{n+1} \geq \alpha_o(1 - \alpha_S^{n+1})$	$\alpha_G^{nf} \geq \alpha_o(1 - \alpha_S^{n+1})$	$\alpha_G^n \geq \alpha_o(1 - \alpha_S^n)$	Comments
1	Yes	Yes	Yes	Nominal two-phase case.
2	Yes	Yes	No	Two-phase: standard case for treating the single-phase to two-phase transition.
3	Yes	No	Yes	Two-phase: difficult to obtain as significant charges in α_S are required. Treated as a nominal two-phase case.
4	No	Yes	Yes	Single-phase; all vaporization/condensation calculated in subroutine PHASE is removed for the pressure iteration.
5	Yes	No	No	Single-phase; pressure and density estimates are initialized to values without convection, and α_G^{n+1} is reset to α_G^{nf} .
6	No	Yes	No	Single-phase analog to case 3; again difficult to obtain. Treated as a nominal single-phase case.
7	No	No	Yes	Single-phase; combination of cases 4 and 5. Pressure and density estimates are reset, α_G is changed, and vaporization/condensation is eliminated.
8	No	No	No	Nominal single-phase case.

4. Single-Phase-Flow Explicit Calculation. Cases 6 and 8 of Table XVI use the nominal single-phase algorithm. The pressure in a single-phase cell is determined from the liquid-field component pressures and compressibilities. First, an EOS call is made with the updated α_G^{n+1} to get T_{Lm}^{n+1} and to obtain estimated values for $\tilde{R}_M$, $\tilde{T}_G$, and $\tilde{G}_L$. Second, the saturation temperatures are estimated from $\tilde{T}_{sat,M} = T_{Lm}^{n+1} - T_{Sup,M}$. Third, the contribution to the liquid field saturation pressure is calculated from the minimum of three terms as

$$\hat{p}_{Lm} = \min\{p_{vM}(\tilde{T}_{Sat,M}) \quad , \quad \tilde{\rho}_{Lm}^n \left[\frac{1}{\tilde{G}_L} - \frac{\alpha_{Lm}}{(1 - \alpha_o)c_{Lm}^2} \right]^{-1} \quad , \quad [(\tilde{\rho}_{Gm}^n + \tilde{\rho}_{Lm}^n)\tilde{R}_M\tilde{T}_G]/\alpha_G^{n+1} \} \quad ,$$

$$m = 1, \dots, \text{NCLE-2} \quad , \quad (\text{IV-150})$$

where $\tilde{\rho}_{Lm}^n = \rho_{Lm}^n - \Gamma_{Lm}\Delta t$.

Fourth, the liquid-field saturation pressure is formed as the sum of the individual $\hat{p}_{Lm}$:

$$\hat{p}_L = \sum_{m=1}^{\text{NCLE-2}} \hat{p}_{Lm} \quad . \quad (\text{IV-151})$$

Fifth, the liquid-field compressibility is adjusted back to the value appropriate to the transition from single- to two-phase flow,

$$\frac{1}{\tilde{G}_L} = \frac{1}{\tilde{G}_L} \left[\frac{(1 - \alpha_S^{n+1})(1 - \alpha_o)}{1 - \alpha_S^{n+1} - \tilde{\alpha}_G^{n+1}} \right] \quad , \quad (\text{IV-152})$$

and the mesh cell pressure is computed to be

$$\tilde{p} = \hat{p}_L + G_L \tilde{\rho}_L \left(\frac{\alpha_{Go}^{n+1} - \tilde{\alpha}_G^{n+1}}{1 - \alpha_S^{n+1} - \tilde{\alpha}_G^{n+1}} \right) \quad , \quad (\text{IV-153})$$

where $\alpha_{G0}^{n+1} = \alpha_0(1 - \alpha_S^{n+1})$ is the vapor volume fraction at the transition between single-phase and two-phase flow.

To ensure a smooth transition from single- to two-phase flow, a vapor state is computed for single-phase mesh cells. First, the saturation temperatures are recalculated if necessary for consistency with the partial pressures given by $\hat{p}_{Lm}$. These saturation temperatures will be called $T_{Sat,M}^{n+1}$. Second, saturated vapor densities are determined. For the analytic equation of state, this means defining a vapor temperature by using Eq. (III-70), where now

$$T_G^{n+1} = \frac{\sum_{m=1}^{NMAT-1} \left(\frac{\hat{p}_{Lm} c_{vGM} T_{Sat,M}^{n+1}}{\tilde{R}_M} \right)}{\sum_{m=1}^{NMAT-1} \left(\frac{\hat{p}_{Lm} c_{vGM}}{\tilde{R}_M} \right)} . \quad (IV-154)$$

Then, the saturated vapor microscopic density is given by solving

$$\rho_{Gm}^{n+1} = \frac{\hat{p}_{Lm}}{\tilde{R}_M T_G^{n+1}} , \quad m = 1, \dots, NMAT-1 . \quad (IV-155)$$

Third, the vapor-field densities are summed to give the total microscopic density for the vapor field, ρ_G^{n+1} . Fourth, $T_{Sat,M}^{n+1}$ allows determination of $e_{vap,M}^{n+1}$ and the vapor-field mixture energy is computed from

$$e_G^{n+1} = \frac{1}{\rho_G^{n+1}} \sum_{m=1}^{NMAT-1} \rho_{Gm}^{n+1} e_{vap,M}^{n+1} . \quad (IV-156)$$

The macroscopic densities for the vapor field are obtained by multiplying the microscopic densities by the vapor-field volume fraction, α_G^{n+1} . For the density components composing the fuel vapor component, the incremental change in vapor density is assigned to the fertile and fissile vapor components, depending on the sign of the change. If the change in vapor density is negative, the loss is assigned to vapor mass fractions as

$$\bar{\rho}_{gm}^{n+1} = \bar{\rho}_{gm}^n + x_{gmG1} (\bar{\rho}_{G1}^{n+1} - \bar{\rho}_{G1}^n) , m = 1,2 \quad . \quad (IV-157)$$

If the change in vapor density is positive, the gain comes from the liquid mass fractions as

$$\bar{\rho}_{gm}^{n+1} = \bar{\rho}_{gm}^n + \bar{x}_{mL1} (\bar{\rho}_{G1}^{n+1} - \bar{\rho}_{G1}^n) , m = 1,2 \quad , \quad (IV-158)$$

where the mass fractions are evaluated using the $\bar{\rho}_{Lm}^n$ from the definition following Eq. (IV-150).

Four final operations are performed to complete the explicit calculation for single-phase flow.

- (1) To force the vapor field to move with the liquid field, the liquid-vapor momentum-exchange function is set to a large number,

$$K_{GL} = 10^{10} \quad . \quad (IV-159)$$

- (2) The vapor-structure momentum-exchange function is set to zero whenever α_S^n is less than the infinite-drag cutoff, ALDRG. In this case,

$$K_{GS} = 0 \quad . \quad (IV-160)$$

- (3) The pressure given by Eq. (IV-153) is merely the first estimate in a pressure iteration scheme to be described in Sec. IV.F. Consistent variation of the initial pressure estimate and liquid density will not change the converged values. Alternatively, extremely large initial pressure gradients can lead to pressure iteration instabilities at a two-phase/single-phase interface. Such difficulties can be avoided by limiting the magnitude of $\tilde{p}$ compared with the previous cycle pressure, p^n . The algorithm is:

If $p^n > \bar{p}_L$, and $\tilde{p} > 2p^n$, then $\tilde{\bar{p}}_L$ is replaced by $[\tilde{\bar{p}}_L - \frac{1}{G_L} (\tilde{p} - 2p^n)]$

and $\tilde{p}$ is set to $2p^n$.

- (4) To obtain mass conservation, the vapor gain or loss is introduced as an additional liquid mass source term that will be incorporated into the component liquid densities before the final implicit liquid density update. These sources are $(\bar{\rho}_{gm}^n - \bar{\rho}_{gm}^{n+1})/\Delta t$. Values also are assigned the GAMCE array (Table XII) so that mass shifts in single-phase cells are accounted for properly in the momentum equations.

5. Two-Phase-Flow Explicit Calculation. Cases 1 and 3 use the nominal two-phase-flow algorithm. First, the fission gas sources described in Sec. IV.E.1 are determined. These include the interfacial term $\Gamma_{g6,extra}$, and any addition to Γ_{g6} resulting from fabricated fuel breakup. The energy per unit mass associated with $\Gamma_{g6,extra}$ is simply e_G^n . Fission gas from fabricated fuel breakup is given the former fabricated fuel temperature, and these energy sources are assigned to Q_G . In addition, the energy loss from the vapor because of the work term associated with fission gas expansion into single phase cells in the previous time step,

$$W_{ex} = \alpha_G^n p^n \Gamma_{g6,extra} / \bar{\rho}_G^n \quad , \quad (IV-161)$$

is calculated.

Second, estimated advanced-time vapor-field densities for cells with two-phase flow are computed from the vapor-field continuity equations,

$$\tilde{\bar{\rho}}_{gm} = \bar{\rho}_{gm}^n - \Delta t \left(\Gamma_{gm} + \frac{\Delta_i <\bar{\rho}_{gm} r_i^{\frac{1}{2}} U_G>_j^n}{r_i^{\frac{1}{2}} \Delta r_i} + \frac{\Delta_j <\bar{\rho}_{gm} v_G>_i^n}{\Delta z_j} \right) \quad ,$$

$$m = 1, \dots, NCGR \quad .$$

$$(IV-162)$$

The estimated densities of the vapor-field density components are summed to give the estimated total macroscopic density for the vapor field,

$$\bar{\rho}_G = \sum_{m=1}^{NCGR} \bar{\rho}_{gm} \quad , \quad (IV-163)$$

from which the estimated mass fractions of the vapor-field density components are computed,

$$\bar{x}_{gm} = \frac{\bar{\rho}_{gm}}{\bar{\rho}_G} \quad . \quad (IV-164)$$

Third, the work term W is evaluated as

$$W = p^n \left[\frac{\Delta_i <\alpha_{Gr}^s U_G>_j^n + \Delta_i <\alpha_{Lr}^s U_L>_j^n}{r_i^s \Delta r_i} + \frac{\Delta_j <\alpha_G^v V_G>_i^n + \Delta_j <\alpha_L^v V_L>_i^n}{\Delta z_j} \right] - 2W_{ex} \quad .$$

(IV-165)

This work term is averaged with the value computed using end-of-time-step quantities (following the pressure iteration) before its addition to the vapor energy equation. Furthermore, to avoid oscillations in interfacial cells, if $2W_{ex}$ changes the sign of W , the work term is zeroed. Fourth, an optional viscous heating term, VIS , can be evaluated as

$$\begin{aligned} VIS = \nu_G \bar{\rho}_G \left[2 \left\{ \left(\frac{\Delta_i U_{Gj}}{\Delta r_i} \right)^2 + \left(\frac{U_{Gi}}{r_i^s} \right)^2 + \left(\frac{\Delta_j V_{Gi}}{\Delta z_j} \right)^2 \right\} - \frac{2}{3} \left(\frac{\Delta_i (r_i^s U_G)_j}{r_i^s \Delta r_i} + \frac{\Delta_j V_{Gi}}{\Delta z_j} \right)^2 \right. \\ \left. + \left(\frac{\Delta_i V_{Gj}}{\Delta r_i} + \frac{\Delta_j U_{Gi}}{\Delta z_j} \right)^2 \right] \quad . \end{aligned} \quad (IV-166)$$

The final terms in this equation are differenced similarly to the terms in the liquid energy equation given by Eqs. (IV-137) and (IV-138). Fifth, the energy per unit volume is updated by addition of the exchange functions, nuclear heating, and the viscous term by

$$(\bar{\rho}_G e_G)^+ = (\bar{\rho}_G e_G)^n + \Delta t (Q_G + N \sum_{m=1}^{NCGR} \bar{\rho}_{gm} Q_{Ngm} + VIS) \quad (IV-167)$$

The values of $(\bar{\rho}_G e_G)^+$ and $W/2$ are retained for the final evaluation of the vapor energy when end-of-time-step velocities are available.

Sixth, estimated advanced-time vapor field energies for cells with two-phase flow are calculated from

$$\bar{e}_G = \frac{1}{\bar{\rho}_G} [(\bar{\rho}_G e_G)^+ - \Delta t (\frac{\Delta_i <\bar{\rho}_G e_G r^S U_G>_j^n}{r_i^S \Delta r_i} + \frac{\Delta_j <\bar{\rho}_G e_G v_G>_i^n}{\Delta z_j})] \quad (IV-168)$$

This allows an estimated pressure and the pressure derivative with respect to density, $\partial p / \partial \bar{\rho}_G$, to be calculated for two-phase cells from the equation of state.

6. Single-Phase/Two-Phase Transition. Case 2 provides the conditions under which a transition from single-phase to two-phase flow takes place. This transition requires a vapor volume fraction above $\alpha_0(1 - \alpha_S^{n+1})$ both before and after the explicit calculation of the convective terms in the liquid-field continuity equations. The calculation initially proceeds as in nominal single-phase flow to the extent that the vapor densities and internal energy are updated from the saturation temperatures from the e_{Lm}^{n+1} . However, in this case, an iteration is performed on Eq. (IV-155) to define a consistent,* updated R_M^{n+1} .

The updated vapor densities are compared with values at the beginning of the time step to obtain source terms, Γ_{gm} , for use in Eq. (IV-162). The negative of these source terms is used to update the liquid Γ_{lm} as in the single-phase liquid case. The updated vapor internal energy is compared with the beginning-of-time-step value to obtain a Q_G for use in Eq. (IV-167). The

*The saturation temperature is used to define which equation set to use for R_M ; this may not be correct if $g_M < 1$. Programming for this latter case is not available in this version of SIMMER-II.

calculation then proceeds as in a nominal two-phase cell with the evaluation of Eqs. (IV-162) through (IV-168). There are two exceptions. First, the liquid-vapor momentum-exchange function is treated as if the flow were single-phase. Second, the work term, W , is set to zero.

7. Other Single-Phase-Flow Cases. The three remaining cases are all cells with single-phase flow. In Case 4, only the explicit update of the vapor volume fraction, α_G^{n+1} , is below the transition value. Such cells are treated as in nominal single-phase flow, but this case motivated programming the subtraction of any vaporization-condensation rates in determining the liquid mass source-term update. Case 5 is the inverse of Case 4, where only α_G^{n+1} is above the transition value. Here, $(\alpha_G^{n+1} - \alpha_G^n)/\Delta t$ can overestimate significantly the rate at which vapor volume is becoming available, and treating such cells with two-phase-flow assumptions may cause overfilling with liquid in the following pressure iteration. For Case 5, the estimated liquid density is made consistent with α_G^{nf} by summing the $\tilde{\rho}_{Lm}^n$ [defined after Eq. (IV-150)] to obtain a new value for $\tilde{\rho}_L$. The vapor fraction α_G^{n+1} is set to α_G^{nf} with estimated cell pressures from the new value of $\tilde{\rho}_L$ and the previously determined values of c_{Lm}^{n+1} . In Case 7, a cell is changing from a two-phase to a single-phase condition but may be overpressurizing, as a two-phase cell does not reflect the consequences of a large increase in the liquid volume fraction until the next cycle. Here, $\tilde{\rho}_{Lm}$ and α_G^{n+1} are redefined as in Case 5, and vaporization-condensation rates are removed as in Case 4.

F. Implicit Calculation

Following the explicit portion of the calculation, the can-wall failure model of Sec. III.D.2 is executed. This determines whether radial motion will be permitted during the coupled solution of the momentum and continuity equations. This solution, called the "pressure-iteration," has been somewhat modified from the form used in the KACHINA computer code. However, the liquid- and vapor-field momentum equations remain coupled to the liquid- or the vapor-field total mass-continuity equation through the liquid or vapor compressibilities, respectively. The mass-continuity equation is determined by whether the flow in the mesh cell is single-phase or two-phase. Because the equation set is nonlinear, the solution must be obtained iteratively.

1. Explicit Portions of the Momentum Equations. The first step in the pressure iteration is to form the explicit part of the liquid- and vapor-field momentum equations. Toward this end, the following quantities are defined.

$$\begin{aligned}
(\bar{p}_G^{U_G})_\lambda &= (\bar{p}_G^{U_G})_\lambda^n - \Delta t \left(\frac{\Delta_\lambda <\bar{p}_G^{U_G} r^\lambda U_G>_j^n}{r_\lambda^\lambda \Delta r_\lambda} + \frac{\Delta_j <\bar{p}_G^{U_G} v_G>_\lambda^n}{\Delta z_j} \right) \\
&+ \Delta t \nu_G \left\{ \frac{\frac{4}{3} \left[\bar{p}_{Gi+1,j}^n \frac{\Delta_{i+1}(r^\lambda U_{Gj}^n)}{r_{i+1}^\lambda \Delta r_{i+1}} - \bar{p}_{Gi,j}^n \frac{\Delta_i(r^\lambda U_{Gj}^n)}{r_i^\lambda \Delta r_i} \right]}{\Delta r_\lambda} \right. \\
&+ \frac{\left[\bar{p}_{G\lambda\theta}^n \frac{\Delta_\theta(U_{G\lambda}^n)}{\Delta z_\theta} - \bar{p}_{G\lambda,j-\frac{1}{2}}^n \frac{\Delta_{j-\frac{1}{2}}(U_{G\lambda}^n)}{\Delta z_{j-\frac{1}{2}}} \right]}{\Delta z_j} \\
&\left. + \frac{\left[\frac{1}{3} \bar{p}_{G\lambda\theta}^n \frac{\Delta_\lambda(v_{G\theta}^n)}{\Delta r_\lambda} - \bar{p}_{G\lambda,j-\frac{1}{2}}^n \frac{\Delta_{j-\frac{1}{2}}(v_{G\lambda}^n)}{\Delta r_\lambda} \right]}{\Delta z_j} \right\}, \tag{IV-169}
\end{aligned}$$

$$\begin{aligned}
(\bar{p}_G^{V_G})_\theta &= (\bar{p}_G^{V_G})_\theta^n - \Delta t \left(\frac{\Delta_i <\bar{p}_G^{V_G} r^\lambda U_G>_\theta^n}{r_i^\lambda \Delta r_i} + \frac{\Delta_\theta <\bar{p}_G^{V_G} v_G>_i^n}{\Delta z_\theta} \right) \\
&+ \Delta t \nu_G \left\{ \frac{\frac{4}{3} \left[\bar{p}_{Gi,j+1}^n \frac{\Delta_{j+1}(v_{Gi}^n)}{\Delta z_{j+1}} - \bar{p}_{Gi,j}^n \frac{\Delta_j(v_{Gi}^n)}{\Delta z_j} \right]}{\Delta z_\theta} \right\}
\end{aligned}$$

$$\begin{aligned}
& + \frac{\left[\bar{\rho}_{G\lambda\theta}^n r_\lambda^{\frac{1}{2}} \frac{\Delta_\lambda(v_{G\theta}^n)}{\Delta r_\lambda} - \bar{\rho}_{Gi-\frac{1}{2},\theta}^n r_{i-\frac{1}{2}}^{\frac{1}{2}} \frac{\Delta_i(v_{G\theta}^n)}{\Delta r_{i-\frac{1}{2}}} \right]}{(r_i^{\frac{1}{2}} \Delta r_i)} \\
& + \frac{\frac{1}{3} \left[\bar{\rho}_{G\lambda\theta}^n r_\lambda^{\frac{1}{2}} \frac{\Delta_\theta(u_{G\lambda}^n)}{\Delta z_\theta} - \bar{\rho}_{Gi-\frac{1}{2},\theta}^n r_{i-\frac{1}{2}}^{\frac{1}{2}} \frac{\Delta_\theta(u_{Gi-\frac{1}{2}}^n)}{\Delta z_\theta} \right]}{(r_i^{\frac{1}{2}} \Delta r_i)} \left. \right\}, \quad (IV-170)
\end{aligned}$$

$$\overline{(\bar{\rho}_L u_L)_\lambda} = (\bar{\rho}_L u_L)_\lambda^n - \Delta t \left(\frac{\Delta_\lambda \langle \bar{\rho}_L u_L r^{\frac{1}{2}} u_L \rangle_j^n}{r_\lambda^{\frac{1}{2}} \Delta r_\lambda} + \frac{\Delta_j \langle \bar{\rho}_L u_L v_L \rangle_\lambda^n}{\Delta z_j} \right)$$

$$+ \Delta t \nu_L \left\{ \frac{\frac{4}{3} \left[\bar{\rho}_{Li+1,j}^n \frac{\Delta_{i+1}(r^{\frac{1}{2}} u_{Lj}^n)}{r_{i+1}^{\frac{1}{2}} \Delta r_{i+1}} - \bar{\rho}_{Lij}^n \frac{\Delta_i(r^{\frac{1}{2}} u_{Lj}^n)}{r_i^{\frac{1}{2}} \Delta r_i} \right]}{\Delta r_\lambda} \right.$$

$$+ \left. \frac{\left[\bar{\rho}_{L\lambda\theta}^n \frac{\Delta_\theta(u_{L\lambda}^n)}{\Delta z_\theta} - \bar{\rho}_{L\lambda,j-\frac{1}{2}}^n \frac{\Delta_{j-\frac{1}{2}}(u_{L\lambda}^n)}{\Delta z_{j-\frac{1}{2}}} \right]}{\Delta z_j} \right\}$$

$$+ \frac{\frac{1}{3} \left[\bar{\rho}_{L\lambda\theta}^n \frac{\Delta_\lambda(v_{L\theta}^n)}{\Delta r_\lambda} - \bar{\rho}_{L\lambda,j-\frac{1}{2}}^n \frac{\Delta_\lambda(v_{Lj-\frac{1}{2}}^n)}{\Delta r_\lambda} \right]}{\Delta z_j} \left. \right\}, \quad (IV-171)$$

$$\begin{aligned}
(\bar{\rho}_L v_L)_\theta &= (\bar{\rho}_L v_L)_\theta^n - \Delta t \left\{ \frac{\Delta_i \langle \bar{\rho}_L v_L r^\xi u_L \rangle_\theta^n}{r_i^\xi \Delta r_i} + \frac{\Delta_\theta \langle \bar{\rho}_L v_L v_L \rangle_i^n}{\Delta z_\theta} \right\} \\
&+ \Delta t \nu_L \left\{ \frac{\frac{4}{3} \left[\bar{\rho}_{Li, j+1}^n \frac{\Delta_{j+1}(v_{Li}^n)}{\Delta z_{j+1}} - \bar{\rho}_{Li j}^n \frac{\Delta_j(v_{Li}^n)}{\Delta z_j} \right]}{\Delta z_\theta} \right. \\
&+ \frac{\left[\bar{\rho}_{L\lambda\theta}^n r_\lambda^\xi \frac{\Delta_\lambda(v_{L\theta}^n)}{\Delta r_\lambda} - \bar{\rho}_{Li-\frac{1}{2}, \theta}^n r_{i-\frac{1}{2}}^\xi \frac{\Delta_{i-\frac{1}{2}}(v_{L\theta}^n)}{\Delta r_{i-\frac{1}{2}}} \right]}{(r_i^\xi \Delta r_i)} \\
&+ \left. \frac{\frac{1}{3} \left[\bar{\rho}_{L\lambda\theta}^n r_\lambda^\xi \frac{\Delta_\theta(u_{L\lambda}^n)}{\Delta z_\theta} - \bar{\rho}_{Li-\frac{1}{2}, \theta}^n r_{i-\frac{1}{2}}^\xi \frac{\Delta_\theta(u_{Li-\frac{1}{2}}^n)}{\Delta z_\theta} \right]}{(r_i^\xi \Delta r_i)} \right\}, \quad (IV-172)
\end{aligned}$$

where

$$\lambda = i + \frac{1}{2},$$

$$\theta = j + \frac{1}{2},$$

$$\Delta_\lambda f = f_{i+1} - f_i,$$

$$\Delta_\theta f = f_{j+1} - f_j,$$

and the terms involving the kinematic viscosities are evaluated only when the optional viscous option is included.

The explicit convection terms in Eqs. (IV-169) through (IV-172) are defined on the momentum cell interfaces. With a variable mesh spacing, the interfaces normal to the momentum direction are centered between the mesh cell interfaces, where the momentum (velocity) is defined. This results in the donor-cell difference terms for momentum convection having a different form for the r and z direction interfaces. Four examples are as follows.

$$\langle \bar{\rho} U_r \zeta U \rangle_{ij} = (r \zeta U)_{ij} \left[\left(\frac{1}{2} + \eta_i \right) (\bar{\rho} U)_{i-\frac{1}{2},j} + \left(\frac{1}{2} - \eta_i \right) (\bar{\rho} U)_{i+\frac{1}{2},j} \right] , \quad (\text{IV-173})$$

where

$$\eta_i = a_0 \operatorname{sign} (U_{ij}) + \frac{b_0 \Delta t}{\Delta r_i} U_{ij} , \quad \text{and}$$

$$U_{ij} = \frac{1}{2} (U_{i+\frac{1}{2},j} + U_{i-\frac{1}{2},j}) ;$$

$$\langle \bar{\rho} UV \rangle_{\lambda, j-\frac{1}{2}} = v_{\lambda, j-\frac{1}{2}} \left[\left(\frac{1}{2} \frac{\Delta z_j}{\Delta z_{j-\frac{1}{2}}} + \eta_{j-\frac{1}{2}} \right) (\bar{\rho} U)_{\lambda, j-1} \right.$$

$$\left. + \left(\frac{1}{2} \frac{\Delta z_{j-1}}{\Delta z_{j-\frac{1}{2}}} - \eta_{j-\frac{1}{2}} \right) (\bar{\rho} U)_{\lambda j} \right] , \quad (\text{IV-174})$$

where

$$n_{j-\frac{1}{2}} = \frac{a_0}{\Delta z_{j-\frac{1}{2}}} \begin{pmatrix} \Delta z_{j-1} & , \text{ if } v_{\lambda, j-\frac{1}{2}} \geq 0 \\ -\Delta z_j & , \text{ if } v_{\lambda, j-\frac{1}{2}} < 0 \end{pmatrix} + \frac{b_0 \Delta t}{\Delta z_{j-\frac{1}{2}}} v_{\lambda, j-\frac{1}{2}} \quad , \quad \text{and}$$

$$v_{\lambda, j-\frac{1}{2}} = \frac{(\Delta r_i v_{i, j-\frac{1}{2}} + \Delta r_{i+1} v_{i+1, j-\frac{1}{2}})}{2\Delta r_\lambda} \quad ;$$

$$\langle \bar{\rho} v r^{\frac{1}{2}} u \rangle_{\lambda, \theta} = (r^{\frac{1}{2}} u)_{\lambda, \theta} \left[\left(\frac{1}{2} \frac{\Delta r_{i+1}}{\Delta r_\lambda} + \eta_\lambda \right) (\bar{\rho} v)_{i, \theta} + \left(\frac{1}{2} \frac{\Delta r_i}{\Delta r_\lambda} - \eta_\lambda \right) (\bar{\rho} v)_{i+1, \theta} \right] \quad ,$$

(IV-175)

where

$$\eta_\lambda = \frac{a_0}{\Delta r_\lambda} \begin{pmatrix} \Delta r_i & , \text{ if } U_{\lambda \theta} \geq 0 \\ -\Delta r_{i+1} & , \text{ if } U_{\lambda \theta} < 0 \end{pmatrix} + \frac{b_0 \Delta t}{\Delta r_\lambda} U_{\lambda \theta} \quad \text{and}$$

$$U_{\lambda \theta} = \frac{\Delta z_j U_{\lambda, j} + \Delta z_{j+1} U_{\lambda, j+1}}{2\Delta z_\theta} \quad ;$$

and

$$\langle \bar{\rho} v v \rangle_{i, j+1} = v_{i, j+1} \left[\left(\frac{1}{2} + \eta_{j+1} \right) (\bar{\rho} v)_{i, j+\frac{1}{2}} + \left(\frac{1}{2} - \eta_{j+1} \right) (\bar{\rho} v)_{i, j+\frac{3}{2}} \right] \quad , \quad \text{(IV-176)}$$

where

$$\eta_{j+1} = \alpha_0 \operatorname{sign}(v_{i, j+1}) + \frac{b_0 \Delta t}{\Delta z_{j+1}} v_{i, j+1} \quad ,$$

$$v_{i,j+1} = \frac{1}{2} (v_{i,j+\frac{1}{2}} + v_{i,j+\frac{3}{2}}) \quad , \quad \text{and}$$

$$\lambda = i + \frac{1}{2} \quad \text{and} \quad \theta = j + \frac{1}{2} \quad .$$

The terms involving the viscous drag only require simple differences between defined velocity positions.

With these definitions, the momentum equations for the liquid and vapor fields take the form

$$\begin{aligned} (\bar{p}_G U_G)_{\lambda}^{n+1} &= \overline{(\bar{p}_G U_G)}_{\lambda} - \alpha_{G\lambda}^{n+1} \frac{\Delta t}{\Delta r_{\lambda}} \Delta_{\lambda} p_j^{n+1} \\ &+ \Delta t K_{GL\lambda} (U_{L\lambda}^{n+1} - U_{G\lambda}^{n+1}) - \Delta t K_{GS\lambda} U_{G\lambda}^{n+1} \\ &+ \Delta t \Gamma_{LG\lambda} U_{L\lambda}^{n+1} - \Delta t \Gamma_{GL\lambda} U_{G\lambda}^{n+1} \quad , \end{aligned} \quad (\text{IV-177})$$

$$\begin{aligned} (\bar{p}_G V_G)_{\theta}^{n+1} &= \overline{(\bar{p}_G V_G)}_{\theta} - \alpha_{G\theta}^{n+1} \frac{\Delta t}{\Delta z_{\theta}} \Delta_{\theta} p_i^{n+1} + \Delta t \bar{p}_{G\theta}^{n+1} g \\ &+ \Delta t K_{GL\theta} (v_{L\theta}^{n+1} - v_{G\theta}^{n+1}) - \Delta t K_{GS\theta} v_{G\theta}^{n+1} \\ &+ \Delta t \Gamma_{LG\theta} v_{L\theta}^{n+1} - \Delta t \Gamma_{GL\theta} v_{G\theta}^{n+1} \quad , \end{aligned} \quad (\text{IV-178})$$

$$\begin{aligned} (\bar{p}_L U_L)_{\lambda}^{n+1} &= \overline{(\bar{p}_L U_L)}_{\lambda} - \alpha_{L\lambda}^{n+1} \frac{\Delta t}{\Delta r_{\lambda}} \Delta_{\lambda} p_j^{n+1} \\ &+ \Delta t K_{GL\lambda} (U_{G\lambda}^{n+1} - U_{L\lambda}^{n+1}) - \Delta t K_{LS\lambda} U_{L\lambda}^{n+1} \\ &+ \Delta t \Gamma_{GL\lambda} U_{G\lambda}^{n+1} - \Delta t (\Gamma_{LG\lambda} + \Gamma_{LS\lambda}) U_{L\lambda}^{n+1} \quad , \quad \text{and} \end{aligned} \quad (\text{IV-179})$$

$$\begin{aligned}
(\bar{\rho}_L v_L)_{\theta}^{n+1} = & \overline{(\bar{\rho}_L v_L)_{\theta}} - \alpha_{L\theta}^{n+1} \frac{\Delta t}{\Delta z_{\theta}} \Delta_{\theta} p_i^{n+1} + \Delta t \bar{p}_{L\theta}^{n+1} g \\
& + \Delta t K_{GL\theta} (v_{G\theta}^{n+1} - v_{L\theta}^{n+1}) - \Delta t K_{LS\theta} v_{L\theta}^{n+1} \\
& + \Delta t \Gamma_{GL\theta} v_{G\theta}^{n+1} - \Delta t (\Gamma_{LG\theta} + \Gamma_{LS\theta}) v_{L\theta}^{n+1} ,
\end{aligned} \tag{IV-180}$$

where

$\lambda = i + 1/2$, with a j subscript implied where necessary and

$\theta = j + 1/2$, with an i subscript implied where necessary.

The mesh cell interfacial values for K_{GL} , K_{GS} , K_{LS} , Γ_{GL} , Γ_{LG} , and Γ_{LS} , which are defined at mesh cell centers, are obtained by averaging them over the momentum cell:

$$K_{\lambda} = \frac{(\Delta r_j K_{ij} + \Delta r_{i+1} K_{i+1,j})}{2\Delta r_{\lambda}} , \tag{IV-181}$$

$$K_{\theta} = \frac{(\Delta z_j K_{ij} + \Delta z_{j+1} K_{i,j+1})}{2\Delta z_{\theta}} , \tag{IV-182}$$

$$\Gamma_{\lambda} = \frac{(\Delta r_j \Gamma_{ij} + \Delta r_{i+1} \Gamma_{i+1,j})}{2\Delta r_{\lambda}} , \text{ and} \tag{IV-183}$$

$$\Gamma_{\theta} = \frac{(\Delta z_j \Gamma_{ij} + \Delta z_{j+1} \Gamma_{i,j+1})}{2\Delta z_{\theta}} . \tag{IV-184}$$

At the location of the input orifice coefficient, Eq. (IV-182) is replaced by Eqs. (III-179) and (III-180) for $K_{LS\theta}$ and $K_{GS\theta}$. The particle jamming restric-

tions (Sec. III.D.5.e) also will result in a direct multiplier on $K_{LS\theta}$ and $K_{LS\lambda}$ when these restrictions are appropriate. The volume fraction coefficients α_G and α_L in the pressure gradient terms are defined on the mesh cell interfaces according to the fractional flow area across the interface where the pressure gradient acts:

$$\alpha_{G\lambda}^{n+1} = \min (\alpha_{Gi,j}^{n+1}, \alpha_{Gi+1,j}^{n+1}) , \quad (IV-185)$$

$$\alpha_{G\theta}^{n+1} = \min (\alpha_{Gi,j}^{n+1}, \alpha_{Gi,j+1}^{n+1}) , \quad (IV-186)$$

$$\alpha_{L\lambda}^{n+1} = 1 - \max (\alpha_{Si,j}^{n+1}, \alpha_{Si+1,j}^{n+1}) - \alpha_{G\lambda}^{n+1} , \text{ and} \quad (IV-187)$$

$$\alpha_{L\theta}^{n+1} = 1 - \max (\alpha_{Si,j}^{n+1}, \alpha_{Si,j+1}^{n+1}) - \alpha_{G\theta}^{n+1} . \quad (IV-188)$$

For the axial interface below where the lower plug interface is located (when the PLUG option is defined), a minimum function is used in Eq. (IV-188). This allows sodium inflow as pictured in Fig. 14. The macroscopic densities $\bar{p}_{G2}$, $\bar{p}_{G\theta}$, $\bar{p}_{L\lambda}$, $\bar{p}_{L\theta}$ defining momentum on the λ and θ interfaces have their values averaged for consistency with the exchange functions, as

$$\bar{p}_{G\lambda} = \frac{(\Delta r_i \bar{p}_{Gi,j} + \Delta r_{i+1} \bar{p}_{Gi+1,j})}{2\Delta r_\lambda} , \quad (IV-189)$$

$$\bar{p}_{G\theta} = \frac{(\Delta z_j \bar{p}_{Gi,j} + \Delta z_{j+1} \bar{p}_{Gi,j+1})}{2\Delta z_\theta} , \quad (IV-190)$$

$$\bar{p}_{L\lambda} = \frac{(\Delta r_i \bar{p}_{Li,j} + \Delta r_{i+1} \bar{p}_{Li+1,j})}{2\Delta r_\lambda} , \text{ and} \quad (\text{IV-191})$$

$$\bar{p}_{L\theta} = \frac{(\Delta z_j \bar{p}_{Li,j} + \Delta z_{j+1} \bar{p}_{Li,j+1})}{2\Delta z_\theta} . \quad (\text{IV-192})$$

2. The Velocity Solution of the Momentum Equations. Having defined the time and space-differenced momentum equations [Eqs. (IV-177) through (IV-180)], we now solve for their velocity components ($U_{G\lambda}$, $V_{G\theta}$, $U_{L\lambda}$, and $V_{L\theta}$). To do this, we separate the momentum density into the product of its macroscopic density and velocity components. Solving the four equations for the four velocity component unknowns gives

$$(\tilde{U}_G)_\lambda = \frac{1}{D_\lambda} \{ [\tilde{\bar{p}}_{L\lambda} + \Delta t(K_{LS\lambda} + K_{GL\lambda} + \Gamma_{LG\lambda} + \Gamma_{LS\lambda})]C_\lambda + \Delta t(K_{GL\lambda} + \Gamma_{LG\lambda})B_\lambda \} , \quad (\text{IV-193})$$

$$(\tilde{V}_G)_\theta = \frac{1}{D_\theta} \{ [\tilde{\bar{p}}_{L\theta} + \Delta t(K_{LS\theta} + K_{GL\theta} + \Gamma_{LG\theta} + \Gamma_{LS\theta})]C_\theta + \Delta t(K_{GL\theta} + \Gamma_{LG\theta})B_\theta \} , \quad (\text{IV-194})$$

$$(\tilde{U}_L)_\lambda = \frac{1}{D_\lambda} \{ [\tilde{\bar{p}}_{G\lambda} + \Delta t(K_{GS\lambda} + K_{GL\lambda} + \Gamma_{GL\lambda})]B_\lambda + \Delta t(K_{GL\lambda} + \Gamma_{GL\lambda})C_\lambda \} , \text{ and} \quad (\text{IV-195})$$

$$(\tilde{v}_L)_\theta = \frac{1}{D_\theta} \{ [\tilde{\rho}_{G\theta} + \Delta t(K_{GS\theta} + K_{GL\theta} + \Gamma_{GL\theta})] B_\theta + \Delta t(K_{GL\theta} + \Gamma_{GL\theta}) C_\theta \} ,$$

(IV-196)

where

$$B_\lambda = \overline{(\tilde{\rho}_L v_L)_\lambda} - \frac{\Delta t}{\Delta r_\lambda} \alpha_{L\lambda}^{n+1} \Delta_\lambda \tilde{p}_j , \quad (IV-197)$$

$$B_\theta = \overline{(\tilde{\rho}_L v_L)_\theta} - \frac{\Delta t}{\Delta z_\theta} \alpha_{L\theta}^{n+1} \Delta_\theta \tilde{p}_i + \Delta t \tilde{\rho}_{L\theta} g , \quad (IV-198)$$

$$C_\lambda = \overline{(\tilde{\rho}_G u_G)_\lambda} - \frac{\Delta t}{\Delta r_\lambda} \alpha_{G\lambda}^{n+1} \Delta_\lambda \tilde{p}_j , \quad (IV-199)$$

$$C_\theta = \overline{(\tilde{\rho}_G v_G)_\theta} - \frac{\Delta t}{\Delta z_\theta} \alpha_{G\theta}^{n+1} \Delta_\theta \tilde{p}_i + \Delta t \tilde{\rho}_{G\theta} g , \quad (IV-200)$$

$$\begin{aligned} D_\lambda = & [\tilde{\rho}_{L\lambda} + \Delta t(K_{LS\lambda} + K_{GL\lambda} + \Gamma_{LG\lambda} + \Gamma_{LS\lambda})] [\tilde{\rho}_{G\lambda} + \Delta t(K_{GS\lambda} + K_{GL\lambda} + \Gamma_{GL\lambda})] \\ & - \Delta t^2 (K_{GL\lambda} + \Gamma_{LG\lambda})(K_{GL\lambda} + \Gamma_{GL\lambda}) , \text{ and} \end{aligned} \quad (IV-201)$$

$$D_\theta = [\tilde{\rho}_{L\theta} + \Delta t(K_{LS\theta} + K_{GL\theta} + \Gamma_{LG\theta} + \Gamma_{LS\theta})][\tilde{\rho}_{G\theta} + \Delta t(K_{GS\theta} + K_{GL\theta} + \Gamma_{GL\theta})] - \Delta t^2(K_{GL\theta} + \Gamma_{LG\theta})(K_{GL\theta} + \Gamma_{GL\theta}) \quad , \quad (IV-202)$$

where

$$\lambda = i + \frac{1}{2}, j \quad \text{and}$$

$$\theta = i, j + \frac{1}{2} \quad .$$

To solve for the advanced-time fluid state, Eqs. (IV-193) through (IV-196) are used to evaluate the advanced-time velocity components on the mesh cell interfaces. However, to evaluate them, we need to know the advanced-time pressure and macroscopic density distributions. Because all three of these quantities are initially unknown, an iterative procedure is required to evaluate them. First, for two-phase cells the pressure distribution is estimated using the vapor-mixture equation of state based on the advanced-time, explicitly evaluated vapor density and energy estimates:

$$\tilde{p}^{\kappa=0} = p(\tilde{\rho}_G, \tilde{e}_G) \quad . \quad (IV-203)$$

For single-phase cells, $\tilde{p}^{\kappa=0}$ comes from Eq. (IV-153) with the possible limit of $2p^n$ as explained in Sec. IV.E.4. The previously evaluated estimates of the advanced-time component macroscopic densities provide initial estimates for the liquid and vapor macroscopic densities:

$$\tilde{\rho}_L^{\kappa=0} = \tilde{\rho}_L \quad \text{for all computational cells, and} \quad (IV-204)$$

$$\tilde{\rho}_G^{\kappa=0} = \tilde{\rho}_G \quad \text{for all two-phase cells.} \quad (IV-205)$$

For single-phase cells, $\tilde{\rho}_G^\kappa$ is given by the summation of the $\tilde{\rho}_{Gm}^{n+1}$, which is obtained by multiplying the microscopic densities from Eq. (IV-155) by α_G^{n+1} for all iterations, κ . With these initial estimates for pressure and macroscopic

density, Eqs. (IV-193) through (IV-196) are solved for initial estimates of the mesh cell interface velocity components:

$$\tilde{U}_{G\lambda}^{\kappa=0}, \tilde{V}_{G\theta}^{\kappa=0}, \tilde{U}_{L\lambda}^{\kappa=0}, \text{ and } \tilde{V}_{L\theta}^{\kappa=0} .$$

The κ superscript is the iteration index with $\kappa = 0$ representing the initial estimate for the unknown quantity.

3. Potential Modification of Two-Phase Cells. One of the problems with the KACHINA-type pressure-iteration is the inability of two-phase cells to anticipate the transition to a single-phase condition resulting from incoming liquid. Cell overfilling can lead to spurious high pressures in the ensuing time step.

In a single-component system, one solution to this problem is given by the algorithm in the K-FIX code.⁵ A consistent solution for a multicomponent system also has been demonstrated.⁶ However, as was stated in Sec. II, SIMMER-II development has concentrated on the treatment of transport processes, and the current version of SIMMER-II cannot transform two-phase cells to single-phase cells during the pressure-iteration.

However, the current SIMMER-II algorithm does attempt to anticipate the overfilling problem. For all two-phase cells where $\alpha_G/(1 - \alpha_S) < 0.75$, an estimated liquid density is computed for the end of the next time step as

$$\bar{\rho}_{L,est}^{n+2} = \tilde{\rho}_L - \Delta t \left[\frac{\Delta_i \langle \tilde{\rho}_L r \xi \tilde{U}_L^{\kappa=0} \rangle_j}{r_i^2 \Delta r_i} + \frac{\Delta_j \langle \tilde{\rho}_L \tilde{V}_L^{\kappa=0} \rangle_i}{\Delta z_j} \right] . \quad (IV-206)$$

A liquid volume fraction is estimated from

$$\alpha_{L,est}^{n+2} = \alpha_L^{n+1} \bar{\rho}_{L,est}^{n+2} / \tilde{\rho}_L , \quad (IV-207)$$

and a vapor volume fraction is estimated from

$$\alpha_{G,est}^{n+2} = 1 - \alpha_{L,est}^{n+2} - \alpha_S^{n+1} . \quad (IV-208)$$

If $\alpha_{G,est}^{n+2} < \alpha_o(1 - \alpha_S^{n+1})$, the cell is anticipated to overfill, and it is treated as a single-phase cell with a pseudo-compressibility. Here G_L is given by

$$\frac{1}{G_L} = \frac{\bar{\rho}_{L,est}^{n+2} - \bar{\rho}_L^n}{\bar{p}_{est}^{n+2} - \bar{p}_{ij}} = \frac{\bar{\rho}_{L,est}^{n+2} - \bar{\rho}_L^n}{\tilde{G}_L \tilde{\rho}_L} \left[\frac{(1 - \alpha_S^{n+1})(1 - \alpha_o)}{\alpha_{Go}^{n+1} - \alpha_{G,est}^{n+2}} \right], \quad (IV-209)$$

where

$\tilde{G}_L$ is calculated from Eq. (IV-145),

$\tilde{\rho}_L^n$ is calculated by summing $\tilde{\rho}_{Lm}^n$ from Eq. (IV-150),

$\bar{p}_{est}^{n+2}$ is the estimated pressure if the liquid density was $\bar{\rho}_{L,est}^{n+2}$,

$\bar{p}_{ij}$ is the current pressure from Eq. (IV-203), and

the final expression is obtained by combining Eqs. (IV-152) and (IV-153) with $\alpha_{G,est}^{n+2}$ substituted for $\tilde{\alpha}_G^{n+1}$, $\bar{p}_{est}^{n+2}$ substituted for $\tilde{p}$, and $\bar{p}_{ij}$ substituted for $\hat{p}_L$. If $\alpha_G^{n+1}/(1 - \alpha_S^{n+1}) > 0.5$, G_L is decreased by an order of magnitude to allow a more rapid change in density with pressure. Following the calculation of G_L , $\tilde{\rho}_L^{\kappa=0}$ is set to $\tilde{\rho}_L^n$, and the liquid and vapor momentum fields are tied together. This means that the K_{GL} 's at the boundaries of this cell are set to 10^{10} .

This anticipation algorithm will not stop a pressure pulse from propagating across a pseudo single-phase cell and overfilling a neighboring two-phase cell where the liquid velocities are small when $\kappa = 0$. This more sophisticated overfilling problem must be controlled by the time-step size as explained in Sec. IV.G.5.d.

4. The Implicit Mass-Continuity Equation. The calculative sequence within the pressure iteration loop begins with an evaluation of the error in the mass-continuity equation in each mesh cell that is caused by the estimated liquid- and vapor-field macroscopic densities and velocities. The mass continuity equation used to obtain the error depends on whether the mesh cell has single-phase or two-phase flow. For single-phase flow mesh cells, the liquid-field total mass continuity equation is used:

$$\tilde{D}_L^\kappa = \frac{1}{\Delta t} [(\tilde{\rho}_L)^\kappa - (\bar{\rho}_L)^n] + \frac{\Delta_i \langle \tilde{\rho}_L^\kappa \tilde{U}_L^\kappa \rangle_j}{r_i^\xi \Delta r_i} + \frac{\Delta_j \langle \tilde{\rho}_L^\kappa \tilde{V}_L^\kappa \rangle_i}{\Delta z_j} + \Gamma_L, \quad (IV-210)$$

where

$\tilde{D}_L^\kappa$ is the total mass-continuity equation error for the liquid field in mesh cell (i,j) and

Γ_L is the liquid-field total mass-transfer rate.

In two-phase flow mesh cells, the vapor-field total mass-continuity equation is used:

$$\tilde{D}_G^\kappa = \frac{1}{\Delta t} [(\tilde{\rho}_G)^\kappa - (\rho_G)^n] + \frac{\Delta_i \langle \tilde{\rho}_G^\kappa \tilde{U}_G^\kappa \rangle_j}{r_i^\dagger \Delta r_i} + \frac{\Delta_j \langle \tilde{\rho}_G^\kappa \tilde{V}_G^\kappa \rangle_i}{\Delta z_j} + \Gamma_G, \quad (\text{IV-211})$$

where

$\tilde{D}_G^\kappa$ is the total mass-continuity equation error for the vapor field in mesh cell (i,j) and

Γ_G is the vapor-field total mass-transfer rate.

5. The Pressure-Change Estimate. In general, the mass-continuity equation residual $\tilde{D}$ is not zero. To force it to zero, the pressure in the mesh cell is changed so that the resultant interface velocities calculated from the momentum equations and the microscopic densities calculated from the equations of state give a zero residual in the next iteration. In the original technique developed for the KACHINA code, only the pressure change in the local mesh cell was considered to change the interface velocities. In SIMMER-II, the pressure changes in the four adjacent mesh cells also are considered to change the interface velocities.

Because the residual $\tilde{D}$ is a function of the local mesh cell pressure and the adjacent mesh cell pressures, $\tilde{D}$ for the " $\kappa+1$ " iteration is expanded in a Taylor series truncated to first order:

$$\tilde{D}_{ij}^{\kappa+1} = \tilde{D}_{ij}^\kappa + \left(\frac{\partial \tilde{D}_{ij}^\kappa}{\partial \tilde{p}_{ij}} \right) \delta \tilde{p}_{ij} + \left(\frac{\partial \tilde{D}_{ij}^\kappa}{\partial \tilde{p}_{i+1,j}} \right) \delta \tilde{p}_{i+1,j} + \left(\frac{\partial \tilde{D}_{ij}^\kappa}{\partial \tilde{p}_{i-1,j}} \right) \delta \tilde{p}_{i-1,j}$$

$$+ \left(\frac{\partial \tilde{D}_{ij}^{\kappa}}{\partial \tilde{p}_{i,j+1}} \right) \delta \tilde{p}_{i,j+1} + \left(\frac{\partial \tilde{D}_{ij}^{\kappa}}{\partial \tilde{p}_{i,j-1}} \right) \delta \tilde{p}_{i,j-1} \quad (IV-212)$$

Because $\tilde{D}_{ij}^{\kappa}$ is known and $\tilde{D}_{ij}^{\kappa+1}$ is required to be zero, Eq. (IV-212) for each of the IBAR*JBAR mesh cells can be reformulated as the matrix equation

$$\underline{\beta}^{\kappa} \cdot \underline{\delta \tilde{p}} = -\underline{\tilde{D}}^{\kappa} \quad (IV-213)$$

of order IBAR*JBAR. The coefficient matrix $\underline{\beta}^{\kappa}$ is derived in Appendix F from its $\partial \tilde{D} / \partial \tilde{p}$ definition in Eq. (IV-212). The sparse five-diagonal, nonzero-element form of $\underline{\beta}^{\kappa}$ is evaluated each iteration using the current iteration estimates of the velocities and densities. The pressure change distribution vector $\underline{\delta \tilde{p}}$ is solved for simultaneously using the SLOR or AIM matrix solution techniques described in Appendix G.

The $\delta \tilde{p}$ pressure change distribution determined from solving matrix Eq. (IV-213) is only an estimate of the actual pressure change needed to make $\underline{\tilde{D}}^{\kappa+1}$ zero. This results from truncating the Taylor series expansion defining $\underline{\tilde{D}}^{\kappa+1}$ in Eq. (IV-213) to first order. The evaluated pressure change is based on a linear extrapolation to the final fluid state. Curvature in the nonlinear fluid-dynamics equations requires several linearly extrapolated pressure-change estimates, based on the local surface tangent, $\underline{\beta}^{\kappa}$, to arrive at the final fluid state solution.

6. Final-State Pressure, Density, and Velocity Estimates. After the evaluation of the pressure change distribution by Eq. (IV-213), the next iteration estimate for the end-of-time step pressure distribution is obtained by

$$\tilde{p}^{\kappa+1} = \tilde{p}^{\kappa} + \delta \tilde{p} \quad (IV-214)$$

The pressure change estimate also is used to evaluate the next estimate for the total field density affecting pressure in each mesh cell.

For cells with single-phase flow, the liquid-field total density is estimated to be

$$\tilde{\rho}_L^{\kappa+1} = \tilde{\rho}_L^{\kappa} + \delta\tilde{p}/G_L \quad , \quad (IV-215)$$

whereas for cells with two-phase flow, the vapor-field total density is determined by

$$\tilde{\rho}_G^{\kappa+1} = \tilde{\rho}_G^{\kappa} + \delta\tilde{p}/G_G \quad . \quad (IV-216)$$

These estimates for pressure and density allow new velocity component estimates to be evaluated using Eqs. (IV-193) through (IV-196). With updated estimates for all three of these unknown final state parameters, the pressure iteration loop returns and evaluates the residual error in the mass-continuity equation. This iteration continues until the error in the mass-continuity equation decreases below a specified convergence criterion. In SIMMER-II, these criteria have the forms

$$|\tilde{D}_{Lij}| \leq \epsilon_{L1} \left\{ \epsilon_{L2} + \max_{i,j} \left[\tilde{\rho}_L^0 \left(\frac{|\tilde{U}_{L,i+\frac{1}{2}}^0| + |\tilde{U}_{L,i-\frac{1}{2}}^0|}{2\Delta r_i} + \frac{|\tilde{V}_{L,j+\frac{1}{2}}^0| + |\tilde{V}_{L,j-\frac{1}{2}}^0|}{2\Delta z_j} \right) \right] \right\} \quad (IV-217)$$

for mesh cells with single-phase flow and

$$|\tilde{D}_{Gij}| \leq \epsilon_{G1} \left\{ \epsilon_{G2} + \max_{i,j} \left[\tilde{\rho}_G^0 \left(\frac{|\tilde{U}_{G,i+\frac{1}{2}}^0| + |\tilde{U}_{G,i-\frac{1}{2}}^0|}{2\Delta r_i} + \frac{|\tilde{V}_{G,j+\frac{1}{2}}^0| + |\tilde{V}_{G,j-\frac{1}{2}}^0|}{2\Delta z_j} \right) \right] \right\} \quad . \quad (IV-218)$$

for mesh cells with two-phase flow. These convergence criteria must be satisfied for all mesh cells before the solution is accepted and the iteration is terminated. However, if the number of pressure iterations exceeds a user-specified maximum, the total calculation is aborted with an appropriate printed comment.

7. Pressure Iteration Summary. Before the pressure iteration is performed, certain quantities must be initialized.

- (1) The explicit terms in the momentum equations are precalculated [Eqs. (IV-169) through (IV-172)].
- (2) The end-of-time-step pressure is estimated based on the vapor mixture equation of state, Eq. (IV-203), or from the single-phase equation of state.
- (3) The end-of-time-step liquid- and vapor-field total macroscopic densities are estimated by evaluating their explicit mass-continuity equations. These are Eq. (IV-126) for the liquid, allowing for any later modifications as explained in Secs. IV.E.4 and IV.F.3, and either Eq. (IV-155) or Eq. (IV-162) for the vapor.
- (4) The liquid- and vapor-field velocity components are estimated based on Eqs. (IV-193) through (IV-196) and the final pressure and density estimates above.
- (5) Any two-phase cells anticipated to overflow are reinitialized as single-phase cells with psuedo-compressibilities as described in Sec. IV.F.3.

Following this initialization, the pressure iteration is performed.

- (6) The mass-continuity equation residual error is evaluated by Eq. (IV-210) or (IV-211).
- (7) The pressure-change distribution estimate is evaluated by Eq. (IV-213).
- (8) From the pressure-change estimate, the next iteration estimates for pressure and density are evaluated by Eqs. (IV-214) through (IV-216).
- (9) The next iteration velocity component estimates are evaluated by Eqs. (IV-193) through (IV-196).
- (10) The calculation returns to Step 6 if the mass-continuity equation residual error does not satisfy the convergence criteria, Eq. (IV-217) or Eq. (IV-218) and the maximum number of pressure iterations is not reached.

8. Liquid-Field and Vapor-Field Component Final Densities. The pressure iteration determines the end-of-time-step pressures, total densities, and velocities over the computing mesh. For consistency, the liquid-field and vapor-field component densities are determined based on these known interface velocities. Because the field velocity components are now available, the mass-continuity equation for density component m can be written as the linear matrix equation

$$\Delta_F^{n+1} \underline{\bar{\rho}_{Fm}^{n+1}} = \underline{(\bar{\rho}_{Fm}^n - \Delta t \Gamma_{Fm}^n)} \quad \begin{array}{l} F = \ell, m = 1, \dots, NCLR \\ F = g, m = 1, \dots, NCGR \end{array} \quad (IV-219)$$

Δ_F^{n+1} is a sparse, five-diagonal matrix that couples each mesh cell to its four neighboring mesh cells. For all liquid components and all vapor components in two-phase cells, the end-of-time-step velocity components for each field F are used to define Δ_F^{n+1} . The five row elements in Δ_F^{n+1} that define the mass-continuity equation for mesh cell i, j are

$$a_{Fij,ij}^{n+1} = 1 + \Delta t \left[\frac{r_{i+\frac{1}{2}}^\xi U_{Fi+\frac{1}{2},j}^{n+1} \left(\frac{1}{2} \frac{\Delta r_{i+1}}{\Delta r_{i+\frac{1}{2}}} + \eta_{i+\frac{1}{2}} \right) - r_{i-\frac{1}{2}}^\xi U_{Fi-\frac{1}{2},j}^{n+1} \left(\frac{1}{2} \frac{\Delta r_{i-1}}{\Delta r_{i-\frac{1}{2}}} - \eta_{i-\frac{1}{2}} \right)}{r_i \Delta r_i} + \frac{v_{Fi,j+\frac{1}{2}}^{n+1} \left(\frac{1}{2} \frac{\Delta z_{j+1}}{\Delta z_{j+\frac{1}{2}}} + \eta_{j+\frac{1}{2}} \right) - v_{Fi,j-\frac{1}{2}}^{n+1} \left(\frac{1}{2} \frac{\Delta z_{j-1}}{\Delta z_{j-\frac{1}{2}}} - \eta_{j-\frac{1}{2}} \right)}{\Delta z_j} \right], \quad (IV-220)$$

$$a_{Fij,i+1j}^{n+1} = \frac{\Delta t \, r_{i+\frac{1}{2}}^\xi U_{Fi+\frac{1}{2},j}^{n+1} \left(\frac{1}{2} \frac{\Delta r_i}{\Delta r_{i+\frac{1}{2}}} - \eta_{i+\frac{1}{2}} \right)}{r_i^\xi \Delta r_i}, \quad (IV-221)$$

$$a_{Fij,i-1j}^{n+1} = \frac{-\Delta t \, r_{i-\frac{1}{2}}^\xi U_{Fi-\frac{1}{2},j}^{n+1} \left(\frac{1}{2} \frac{\Delta r_i}{\Delta r_{i-\frac{1}{2}}} + \eta_{i-\frac{1}{2}} \right)}{r_i^\xi \Delta r_i}, \quad (IV-222)$$

$$a_{Fij,ij+1}^{n+1} = \frac{\Delta t v_{Fi,j+\frac{1}{2}}^{n+1} \left(\frac{1}{2} \frac{\Delta z_j}{\Delta z_{j+\frac{1}{2}}} - \eta_{j+\frac{1}{2}} \right)}{\Delta z_j}, \text{ and} \quad (\text{IV-223})$$

$$a_{Fij,ij-1}^{n+1} = \frac{-\Delta t v_{Fi,j-\frac{1}{2}}^{n+1} \left(\frac{1}{2} \frac{\Delta z_j}{\Delta z_{j-\frac{1}{2}}} + \eta_{j-\frac{1}{2}} \right)}{\Delta z_j}, \quad (\text{IV-224})$$

where $\eta_{i+1/2}$ is defined by Eq. (IV-130) and the other η quantities are defined similarly. For the vapor components in single-phase cells, the $\bar{p}_{gm}^{n+1}$ are maintained as the values previously determined by setting $a_{gij,ij}^{n+1}$ to unity and the off-diagonal matrix element to zero. Matrix Eq. (IV-219) is solved simultaneously for $\bar{p}_{Fm}^{n+1}$ using the SLOR or AIM method described in Appendix G. This evaluation is performed for all of the eight liquid density components and six vapor density components present. The liquid- and vapor-field total macroscopic densities evaluated in the pressure iteration then are replaced by the sum of their component macroscopic densities.

9. Vapor-Field Final Energy. Because the vapor composition may change significantly during a time step, identical velocities are used to evaluate both the vapor densities and the vapor internal energy.

The work term evaluation is completed with an expression similar to Eq. (IV-165) but that uses the advanced-time velocities, pressures, and volume fractions. This expression is

$$\hat{W} = p^{n+1} \left(\frac{\Delta_i <\alpha_G r^{\$U_G>_j^{n+1} + \Delta_i <\alpha_L r^{\$U_L>_j^{n+1}}}{r_i^{\$ \Delta r_i} + \frac{\Delta_j <\alpha_G v_G>_i^{n+1} + \Delta_j <\alpha_L v_L>_i^{n+1}}{\Delta z_j} \right). \quad (\text{IV-225})$$

The effective work term now is evaluated as

$$W_{\text{eff}} = \frac{1}{2} [W + \hat{W}] \quad \text{if } W \cdot \hat{W} > 0 \quad , \text{ or}$$

$$W_{\text{eff}} = 0 \quad \text{if } W \cdot \hat{W} \leq 0 \quad . \quad (\text{IV-226})$$

Consequently, the work term is eliminated if velocity oscillations cause W and $\hat{W}$ to have different signs.

The vapor internal energy per unit volume is updated using the matrix equation,

$$\Delta_G^{n+1} \frac{(\bar{\rho}_G e_G)^{n+1}}{\sum_{m=1}^{N_{\text{CGR}}} \bar{\rho}_{gm}^{n+1}} = ((\bar{\rho}_G e_G)^+ - \Delta t W_{\text{eff}}) \quad , \quad (\text{IV-227})$$

and the updated vapor specific internal energy is obtained from

$$e_G^{n+1} = \frac{(\bar{\rho}_G e_G)^{n+1}}{\sum_{m=1}^{N_{\text{CGR}}} \bar{\rho}_{gm}^{n+1}} \quad . \quad (\text{IV-228})$$

For single-phase cells, Eq. (IV-227) is modified so that Eq. (IV-228) will reproduce the vapor energies of Eq. (IV-156). This allows consistency to be maintained with the vapor densities determined from Eq. (IV-155).

G. Boundary Conditions and Fluid-Dynamics Controls

The SIMMER-II computing geometry is either a (x,y) plane ($\xi = 0$) or a right circular cylinder ($\xi = 1$). For the planar case with IVIS = 1 (using the optional viscous formulation), the left boundary is treated as rigid with no slip. For all other cases, the left boundary has rigid free-slip characteristics. The other three boundaries (the bottom, top, and right boundaries) are allowed to have one of four types of conditions: a rigid, free-slip boundary condition; a boundary pressure as a function of time; a rigid, no-slip boundary condition; or a continuative outflow boundary condition. As described by the input data (Appendix Q), the right-hand boundary can be divided further into regions described by JRGID.

These four boundary conditions available in SIMMER-II are described below for the bottom boundary. Similar treatments exist for the top and right boundary. For ease in implementing the boundary conditions, an extra row or column of mesh cells is included around the outside of the computing mesh. These "fictitious" mesh cells provide exterior locations for the storage of boundary values that can be used automatically to evaluate the field equations.

1. Rigid, Free-Slip Boundary Condition. A rigid, free-slip boundary condition is used to model an axis, plane of symmetry, or nonadhering surface that exerts no drag on the fluid. For this boundary condition, the normal component of the velocity is zero at the wall, and the gradients of the mesh cell scalar quantities are zero. Thus, for the bottom boundary,

$$(V_L)_{i,j-\frac{1}{2}} = 0 , \quad (IV-229)$$

$$(V_G)_{i,j-\frac{1}{2}} = 0 , \quad (IV-230)$$

$$(p)_{i,j-1} = (p)_{i,j} , \quad (IV-231)$$

$$(\alpha_G)_{i,j-1} = (\alpha_G)_{i,j} , \quad (IV-232)$$

$$(\alpha_L)_{i,j-1} = (\alpha_L)_{i,j} , \quad (IV-233)$$

$$(\alpha_S)_{i,j-1} = (\alpha_S)_{i,j} , \quad (IV-234)$$

$$(\bar{\rho}_{gm})_{i,j-1} = (\bar{\rho}_{gm})_{i,j} , \quad m = 1, \dots, NCGR , \quad (IV-235)$$

$$(e_G)_{i,j-1} = (e_G)_{i,j} , \quad (IV-236)$$

$$(\bar{\rho}_{Lm})_{i,j-1} = (\bar{\rho}_{Lm})_{i,j} , \quad m = 1, \dots, NCLR , \quad (IV-237)$$

$$(e_{Lm})_{i,j-1} = (e_{Lm})_{i,j} , \quad m = 1, \dots, NCLE , \quad (IV-238)$$

$$(\bar{\rho}_{sm})_{i,j-1} = (\bar{\rho}_{sm})_{i,j} , \quad m = 1, \dots, NCSR , \text{ and } \quad (IV-239)$$

$$(e_{Sm})_{i,j-1} = (e_{Sm})_{i,j} , \quad m = 1, \dots, NCSE . \quad (IV-240)$$

In addition, the gradients of the tangential velocity components are zero such that

$$(U_L)_{i+\frac{1}{2},j-1} = (U_L)_{i+\frac{1}{2},j} , \text{ and } \quad (IV-241)$$

$$(U_G)_{i+\frac{1}{2},j-1} = (U_G)_{i+\frac{1}{2},j} , \quad i = 0, 1, \dots, IBAR . \quad (IV-242)$$

2. Boundary Pressure as a Function of Time. Various surfaces in reactor simulation problems, such as the boundaries in a single-subassembly problem, are best modeled by specifying a boundary pressure. In SIMMER-II, the condition has been implemented by modifying Eqs. (IV-229) to (IV-231) of the rigid free-slip boundary conditions. Now $(V_L)_{i,j-1/2}$ and $(V_G)_{i,j-1/2}$ are computed assuming that $P_{i,j-1}$ is input. Because the remaining characteristics of the bottom cell are reflected into the fictitious cell, care must be taken so that reversal of velocities from outflow to inflow does not produce unacceptable inconsistencies, that is, densities and internal energies unacceptably incompatible with the pressure in the fictitious cell.

3. Rigid, No-Slip Boundary Condition. The rigid, no-slip boundary condition can be applied when the explicit calculation of the stress tensor is included in SIMMER-II. It can be applied at the right boundary when $\xi = 1$ or at both left and right boundaries if $\xi = 0$. The boundary conditions are the same as the rigid, free-slip case except that the tangential velocity components also are assumed to be zero at the wall. To force this at the bottom boundary,

$$(U_L)_{i+\frac{1}{2},j-1} = -(U_L)_{i+\frac{1}{2},j} \quad , \quad \text{and} \quad (IV-243)$$

$$(U_G)_{i+\frac{1}{2},j-1} = -(U_G)_{i+\frac{1}{2},j} \quad , \quad i = 0, 1, \dots, IBAR \quad . \quad (IV-244)$$

4. Continuitive Outflow (or Inflow) Boundary Condition. The continuative outflow boundary condition allows the fluid to leave (or enter) the computing mesh at its own calculated rate. For this boundary condition, all gradients are set to zero. Hence, the conditions detailed for the rigid, free-slip boundary are used except that the gradient of the normal velocity component is set to zero; for example, at the bottom boundary,

$$(V_L)_{i,j-\frac{1}{2}} = (V_L)_{i,j+\frac{1}{2}} \quad , \quad \text{and} \quad (IV-245)$$

$$(V_G)_{i,j-\frac{1}{2}} = (V_G)_{i,j+\frac{1}{2}} \quad . \quad (IV-246)$$

In the current version of SIMMER-II, pressures are equilibrated after each iteration when this boundary condition is applied. This appears to be incorrect,⁷ and care should be used in applying this boundary condition option.

5. Fluid-Dynamics Controls. There are 13 controls that influence the fluid-dynamics time-step size in SIMMER-II. In addition, this section discusses situations that will result in problem termination.

a. Initial Time Step, DTL(1). In many calculations for which the initial conditions are far from equilibrium, the SIMMER-II fluid-dynamics calculation performs more smoothly if the first several cycles use a relatively small time step (10^{-6} - 10^{-5} s). It is also convenient (but possibly not efficient) to perform some calculations using a constant time step. Therefore, the initial time step can be maintained for an input number of cycles, NDIO, unless some other time-step control requires a smaller time step.

b. Long Edit, DTL(2). This control ensures that the time specified by the input quantities DTO and DTDC for a long edit of the fluid-dynamics mesh cell variables coincides with the end of a time step. If the time step determined by this control is less than the minimum allowed time step, the minimum time step is used instead.

c. Courant Condition on Material Velocity, DTL(3). Because of the nature of the pressure iteration, SIMMER-II is not restricted by the classic Courant stability condition of the form

$$\frac{c\Delta t}{\Delta x} < 1 \quad ,$$

where c is the sound speed, and Δx is the minimum mesh cell dimension.

This limitation arises in purely explicit treatments of the fluid-dynamics equations. There pressure waves could travel farther than one mesh cell for a large time step, whereas the fluid-dynamics method would allow the wave to travel one mesh cell at most. However, because convection in the momentum equation in SIMMER-II is treated explicitly, the fluid-dynamics method is restricted to convect material one mesh cell in a single time step although pressure waves can be propagated throughout the mesh. This leads to a stability criterion where a trial time-step size for each radial and axial velocity is computed as

$$\Delta t_{ri} = \frac{\min [(r_i^{\frac{1}{2}} \Delta r_i), (r_{i+1}^{\frac{1}{2}} \Delta r_{i+1})]}{[r^{\frac{1}{2}} \max (|U_G|, |U_L|)]_{i+\frac{1}{2}}} , \text{ and} \quad (\text{IV-247})$$

$$\Delta t_{zj} = \frac{\min [\Delta z_j, \Delta z_{j+1}]}{[\max (|V_G|, |V_L|)]_{j+\frac{1}{2}}} . \quad (\text{IV-248})$$

The fluid-dynamics time step then is limited such that

$$\Delta t_t^{n+1} < C_t \min [\Delta t_{ri}, \Delta t_{zj}] ,$$

where C_t is the Courant input constant, $0 < C_t \leq 1.0$.

For stability, C_t should be selected as less than one-half. Otherwise, vaporization-condensation processes can lead to oscillations because the SIMMER-II exchange functions are evaluated without the convective terms in the continuity equations. In the absence of other information, a value of 0.3 is suggested for C_t .

d. Maximum Vapor Volume Fraction Change, DTL(4). In calculations in which two-phase and single-phase flow regimes exist simultaneously, it is possible for a mesh cell to change rapidly from two-phase to single-phase flow. If the change is too rapid, a mesh cell can overfill in a single time step and produce an extremely high single-phase pressure. For cells with a decreasing α_G , DTL(4) is the minimum Δt obtained in three computations: one based on the changes in α_G during the explicit update of cell volume fractions, the second based on anticipated results from the pressure iteration, and the third based on the changes occurring during the pressure iteration. The computations based on the explicit α_G update involve the minimum Δt from two expressions. The first expression is

$$\Delta t^{n+1} = \min_{i,j} \left[\frac{8 \times 10^5 G_{Lo}^{-1} \alpha_G^{n+1}}{\max (\alpha_G^n - \alpha_G^{n+1}, \alpha_L^{n+1} - \alpha_L^n)} \right] \Delta t^n , \quad (\text{IV-249})$$

where G_{Lo}^{-1} is the liquid compressibility, $\partial \rho_L / \partial p$, for pure liquid ($\alpha_L = 1$), based on the densities from Eq. (IV-126).

The second expression is

$$\Delta t^{n+1} = \min_{i,j} \left\{ \frac{[\alpha_G^{n+1} - 0.95 \alpha_o (1 - \alpha_S^{n+1})]}{(\alpha_G^n - \alpha_G^{n+1})} \right\} \Delta t^n . \quad (IV-250)$$

The second expression is evaluated only for two-phase cells. If $\alpha_G^n < 0.75 \alpha_o (1 - \alpha_S^n)$, neither computation is performed. The minimum Δt is limited by $\min (\Delta r_i, \Delta z_j) / (1/2 c_{Lij})$, where c_{Lij} is the liquid sonic velocity and Δr_i and Δz_j are local cell dimensions.

The second calculation examines the $\alpha_{G,est}^{n+2}$ calculated by Eq. (IV-208) at the beginning of the pressure iteration. If $\alpha_{G,est}^{n+2}$ is anticipated to be negative for a two-phase cell, Δt^{n+1} can be no larger than Δt^n .

The computation done after the pressure iteration involves the change in vapor volume fraction, $\Delta \alpha_G$, resulting from the liquid density change during the pressure iteration and applies only to two-phase cells. Again, the minimum Δt from two expressions is determined. The two expressions are

$$\Delta t^{n+1} = \min_{i,j} \left(8 \times 10^5 \frac{G_{Lo}^{-1} \alpha_G^{n+1}}{\Delta \alpha_G} \right) \Delta t^n \quad \text{and} \quad (IV-251)$$

$$\Delta t^{n+1} = \min_{i,j} \left(\frac{\left\{ \frac{1}{2} [\alpha_G^{n+1} - \alpha_o (1 - \alpha_S^{n+1})] + \frac{p^{n+1}}{\rho_L^{n+1}} \frac{\partial \rho_L}{\partial p} \right\}}{\Delta \alpha_G} \right) \Delta t^n . \quad (IV-252)$$

where

$\frac{\partial \rho_L}{\partial p}$ is the liquid compressibility at $\alpha_L = \alpha_L^{n+1}$ and

$\Delta\alpha_G$ is evaluated so that the decreasing vapor volume fraction leads to a positive number.

For this evaluation, Δt is limited by the $\min(\Delta r_i, \Delta z_j)/c_{Lij}$. Finally, if the pressure iteration predicts $\alpha_G^{n+1} < -\alpha_0$ in a supposedly two-phase cell, the problem is aborted to avoid errors in further calculations.

e. Maximum Time-Step Increase, DTL(5). The fluid-dynamics time step is not permitted to increase to more than twice the previous time step. This prevents sudden increases in the time step based on other time-step controls following a rapid transient. Changing the time step by a large amount can increase the error in the solution from time differencing greatly. However, as a consequence of efficiency concerns, Δt^{n+1} is limited by $2\Delta t^{n-1}$ instead of $2\Delta t^n$ in the case of a time-step reduction caused by DTL(2), the long edit.

f. Maximum Time Step, DTL(6). The time step is not permitted to exceed a user-specified maximum value DTMAX.

g. Optimum Pressure Iteration, DTL(7). The time-step size affects the number of pressure iterations required to converge to the end-of-time-step fluid state. An increase in the time-step size increases the number of pressure iterations. From four to six pressure iterations each time step minimize the overall calculative effort of evaluating exchange functions, explicit equations, and implicit equations (requiring a pressure iteration) for typical problems. Time-step control DTL(7) is used to estimate the time-step size that yields a user-specified optimum number of pressure iterations, ITOPT (four to six iterations). If the last time step Δt^n required NUMIT pressure iterations, DTL(7) will be estimated by assuming the time-step size is proportional to the number of pressure iterations,

$$\Delta t^{n+1} = \left(\frac{ITOPT}{NUMIT} \right) \Delta t^n \quad . \quad (IV-253)$$

h. Phase-Transition-Rate Limits, DTL(8). The methods for the phase-transition models ignore the convection terms in the conservation equations. If excessive density changes occur, negative values can be calculated to update internal energies. To avoid such problems, the time step is controlled by phase-transition rates. For vaporization, the maximum permitted time step is limited to

$$\Delta t^{n+1} = \text{Min}_{m,i,j} \left[\frac{(\text{DROJAY}) \bar{p}_G}{\Gamma_{\ell m g m} - \Gamma_{g m \ell m}} \right], \quad (\text{IV-254})$$

where DROJAY is an input quantity defined as the maximum fractional increase in the existing vapor density because of vaporization of any component.

In a similar manner, the condensation rate controls the time step. This control has the form

$$\Delta t^{n+1} = \text{Min}_{m,i,j} \left\{ \left[\frac{(\text{DJMAX}) \bar{p}_G \psi_p}{\Gamma_{g m \ell m} - \Gamma_{\ell m g m}} \right] \left(\frac{\bar{p}_G}{\bar{p}_{gm}} \right) \right\}, \quad (\text{IV-255})$$

where DJMAX is an input quantity defined as the fractional decrease in the existing vapor density because of condensation of any component.

The density ratio in Eq. (IV-255) permits the time step to remain large if density component m constitutes a small fraction of the total existing vapor mass. The quantity ψ_p is an internal multiplier. If the total fraction of vapor condensed is greater than DJMAX and convective outflow is occurring, ψ_p is set to one. Otherwise ψ_p is set to two. DJMAX should be no greater than one-half.

An additional time-step control based on the phase-transition rates is given by

$$\Delta t = \text{Min}_{i,j} \left[\frac{(\text{DJMAX}) \bar{p}_L}{\Gamma_{LG} - \Gamma_{GL}} \right], \text{ provided } \bar{p}_L \geq 10^{-2} \text{ kg m}^{-3}. \quad (\text{IV-256})$$

This control limits the percentage of the liquid-field mass that can be vaporized in a single time step.

i. High Reactivity Time-Step Limitation, DTL(9). The neutronics time-step controls for the quasistatic method are based on current conditions and the previous transient history. Coupling between the neutronics and hydrodynamics is explicit. This approach may allow a large a hydrodynamics time-step to be

taken at high net reactivity, inducing an unacceptable power history. Above an input reactivity (typically 0.90\$), DTL(9) does not permit the time step to exceed an input limit (typically 0.2 ms).

For the point-kinetics option, DTL(9) is controlled by the input variable DPOWMX, the maximum fractional change in the normalized power during a hydrodynamics time step. The actual form for this time-step control is given by Eq. (V-126).

j. Neutronics Time-Step Control DTL(10). This control ensures that the end of a fluid-dynamics time step coincides with the time specified by the quasi-static neutronics time-step algorithm for reactivity or flux shape updates.

For the point-kinetics option, DTL(10) reduces the time-step size as prompt critical is approached. The input variable DREAMX, the maximum reactivity change in a time step, is applied as given in Eq. (V-127).

k. Vapor Specific Internal Energy Change, DTL(11). To ensure gradual changes in the vapor internal energy so that extreme temperature changes do not affect accuracy or stability adversely, the length of the time step can be reduced following calculation of $\bar{e}_G^{n+1}$ based on the exchange functions. This estimated vapor specific internal energy is

$$\bar{e}_G^{n+1} = \frac{1}{\hat{\rho}_G} [\bar{\rho}_G^n \bar{e}_G^n + \Delta \tilde{t} Q_G] \quad , \quad (\text{IV-257})$$

where

$\bar{e}_G^{n+1}$ is the estimated vapor mixture specific internal energy for the time $\tilde{t}^{n+1}$,

$\hat{\rho}_G = \bar{\rho}_G^n - \Delta \tilde{t} \Gamma_G$ is the estimated vapor mixture macroscopic density for the time $\tilde{t}^{n+1}$, and

$\tilde{t}^{n+1}$ is the estimated time using the time step estimate $\Delta \tilde{t}$ obtained before the exchange-function calculation.

The estimated vapor-field specific internal energy is used in two ways to limit the time step. First, if the vapor-energy is decreasing, a 50% change is the largest desired change in the vapor-field specific internal energy. Thus, the time-step control is

$$\Delta t^{n+1} = \text{Min}_{i,j} \left(\frac{e_G^n}{2 |\bar{e}_G^{n+1} - e_G^n|} \right) \Delta \tilde{t} \quad , \text{ for } \bar{e}_G^{n+1} < e_G^n \quad . \quad (\text{IV-258})$$

Second, the maximum pressure change in a mesh cell resulting from the estimated energy change is limited to the maximum of (1) the pressure difference between the pressure of the mesh cell and the adjacent mesh cells, (2) the pressure in the mesh cell, and (3) a pressure of 0.1 MPa. This pressure maximum is denoted by $\Delta p_{\max}$. The time step is limited to

$$\Delta t = \text{Min}_{i,j} \left[\left(\frac{\partial e_G}{\partial p} \right) \frac{\Delta p_{\max}}{|\bar{e}_G^{n+1} - e_G^n|} \right] \Delta \tilde{t} \quad , \quad (\text{IV-259})$$

where the derivative $(\partial e_G / \partial p)$ is estimated from the vapor equation of state. For the analytic equation of state, this derivative is estimated as

$$\frac{\partial e_G}{\partial p} = \frac{\alpha_G \bar{c}_{vG}}{\text{NMAT} \sum_{m=1} \bar{p}_{Gm} R_M} \quad . \quad (\text{IV-260})$$

1. Liquid Specific Internal Energy Change, DTL(12). The energy change in the liquid also is allowed to affect the time-step size. Here, the size of the time step is limited such that the expected fractional change in the specific internal energy does not exceed a fraction DTMPM specified by the user. Thus, for each liquid-field energy component, the time step is limited such that

$$\Delta t^{n+1} = \text{Min}_{m,i,j} \left[\frac{\text{Max} (e_{Lm}^n, e_{Lm}^{n+1}) (\text{DTMPM})}{|e_{Lm}^{n+1} - e_{Lm}^n|} \right] \Delta t^n \quad , \quad (\text{IV-261})$$

where

Δt^n is the time step between times t^n and t^{n+1} and

e_{Lm} is the specific internal energy of liquid-field energy component m.

However, because rapid changes in liquid energy can occur for small densities, the above time-step control is only allowed to be active if

$$e_{Lm}^n > 10^5 \text{ J/kg} ,$$

$$e_{Lm}^{n+1} > 10^5 \text{ J/kg} , \text{ and}$$

$$\bar{\rho}_{Lm}^{n+1} \geq 10^{-3} \text{ kg m}^{-3} .$$

m. Restart Dumps, DTL(13). This control ensures that the time specified by the input quantities DTDM and TDMP for a restart dump coincides with the end of a time step. If the time step determined by this control is less than the minimum allowed time step, the minimum time step is used instead.

n. Problem Termination. The fluid-dynamics calculation in SIMMER-II may be terminated by any one of six controls. Two of the terminations are normal terminations because of depletion of allotted computer time. A third possibility is achieving the maximum number of postprocessor files allowed on TAPE36. A restart capability is provided to continue such cases. The other three terminations are results of problem aborts. The conditions for these aborts are (1) exceeding the maximum number of pressure iterations without convergence, (2) decreasing the time step below an input minimum, DTMIN, and (3) overfilling a two-phase cell in IMFLUD such that $\alpha_G < -\alpha_0$.

H. Tabular Equation-of-State Interpolation Methods

A special "rational function" algorithm has been developed for use with SESAME equation-of-state tables.⁸ In SIMMER-II, this algorithm is used for functions of a single variable. For functions of two independent variables, the user can choose either the rational function method or a bilinear scheme. Selection of the bilinear option may save computing time, but it is not accurate and should be used with caution. A brief outline of the rational function method follows. For more details and examples, the reader is referred to Ref. 8.

Although many methods of interpolation are presented in the literature, thermodynamic quantities present special interpolation problems because they are not smooth functions when phase transitions are present. Standard methods such as spline and Lagrangian interpolation produce nonphysical oscillations in the area of phase transitions. The interpolated surface can have regions of thermodynamic instability resulting in imaginary sound speeds and negative heat capacities.

The rational function method has been found to be useful in dealing with these problems. Consider a function $f(x)$, tabulated at N points, $x_1 < x_2 < \dots < x_N$. Define the quantities

$$\begin{aligned}\Delta_{i+\frac{1}{2}} x &= x_{i+1} - x_i, \\ S_i &= \frac{(f_{i+1} - f_i)}{\Delta_{i+\frac{1}{2}} x}, \text{ and} \\ t_i &= \left(\frac{df}{dx}\right)_i.\end{aligned}\tag{IV-262}$$

A simple rule for estimating the derivative t_i is to fit a quadratic polynomial through the points f_{i-1} , f_i , and f_{i+1} . The result is

$$t_i = \frac{[(\Delta_{i-\frac{1}{2}} x) S_i + (\Delta_{i+\frac{1}{2}} x) S_{i-1}]}{(\Delta_{i-\frac{1}{2}} x + \Delta_{i+\frac{1}{2}} x)}.\tag{IV-263}$$

To estimate the function in the interval $x_i < x < x_{i+1}$, many methods use a cubic polynomial,

$$f(x) = \sum_{k=0}^3 A_k (x - x_i)^k,\tag{IV-264}$$

where the four coefficients A_k are determined from the four quantities f_i , f_{i+1} , t_i , and t_{i+1} . However, such a procedure can result in the nonphysical oscilla-

tions mentioned above. In the SESAME algorithm, the cubic polynomial is replaced by a "rational function,"

$$f(x) = f_i + S_i (x - x_i) - \frac{[(x - x_i)(x_{i+1} - x)(C_1\mu_1 + C_2\mu_2)]}{(\mu_1 + \mu_2) \Delta_{i+\frac{1}{2}} x}, \quad (\text{IV-265})$$

where

$$C_1 = S_i - t_i ,$$

$$C_2 = t_{i+1} - S_i ,$$

$$\mu_1 = C_2 (x_{i+1} - x) , \text{ and}$$

$$\mu_2 = C_1 (x - x_i) .$$

In Ref. 8, the rational function is shown to eliminate the instabilities inherent with the polynomial. This method is particularly useful for functions having a rapid or discontinuous change in slope. However, it is also very accurate for smooth functions.

Next, consider a function $F(x,y)$ tabulated on the mesh, $x_1 < x_2 < \dots < x_N$, $y_1 < y_2 < \dots < y_M$. To estimate $F(x,y)$ on the interval $x_i \leq x \leq x_{i+1}$, $y_i \leq y \leq y_{j+1}$, the following procedure is used. First compute the four quantities

$$f_i = F(x_i, y) ,$$

$$f_{i+1} = F(x_{i+1}, y) ,$$

$$f_j = F(x, y_j) , \quad \text{and}$$

$$f_{j+1} = F(x, y_{j+1}) . \quad (\text{IV-266})$$

By applying the rational-function algorithm, Eq. (IV-265), $F(x,y)$ is estimated from the following expression.

$$\begin{aligned}
F(x,y) = & f_i q_{xi} + f_{i+1}(1 - q_{xi}) + f_j q_{yj} + f_{j+1}(1 - q_{yj}) \\
& - F(x_i, y_j)(1 - q_{xi})(1 - q_{yj}) \\
& - F(x_{i+1}, y_j) q_{xi}(1 - q_{yj}) - F(x_i, y_{j+1})(1 - q_{xi}) q_{yj} \\
& - F(x_{i+1}, y_{j+1}) q_{xi} q_{yj} \quad ,
\end{aligned}
\tag{IV-267}$$

where

$$q_{xi} = \frac{(x - x_i)}{(x_{i+1} - x_i)} \quad \text{and}$$

$$q_{yj} = \frac{(y - y_j)}{(y_{j+1} - y_j)} \quad .$$

This procedure is reasonably fast and accurate. Although it gives discontinuities in the derivatives of the function $F(x,y)$, it does not seem to cause any problems in SIMMER-II.

REFERENCES

1. F. H. Harlow and A. A. Amsden, "Numerical Calculation of Multiphase Fluid Flow," J. Comp. Phys. 17, 19-52 (1975).
2. A. A. Amsden and F. H. Harlow, "KACHINA: An Eulerian Computer Program for Multifield Fluid Flows," Los Alamos Scientific Laboratory report LA-5680 (December 1974).
3. J. E. Cahalan and D. R. Ferguson, "A Preliminary User's Guide to Version 1.0 of the SAS3D LMFBR Accident Analysis Computer Code," Argonne National Laboratory unnumbered report (July 1977).
4. P. Henrici, Elements of Numerical Analysis (John Wiley and Sons, Inc., New York, 1964), pp 91-92.
5. W. C. Rivard and M. D. Torrey, "K-Fix: A Computer program for Transient, Two-Dimensional, Two-Fluid Flow," Los Alamos Scientific Laboratory report LA-NUREG-6623 (April 1977).
6. W. R. Bohl, "A Practical Prescription for Solving the Multiphase Fluid Dynamics Equations," Trans. Am. Nuc. Soc. 39, 502-503 (1981).

7. B. D. Nichols, C. W. Hirt, R. S. Hotchkiss, "SOLA-VOF: A Solution Algorithm for Transient Fluid Flow with Multiple Free Boundaries," Los Alamos National Laboratory report LA-8355 (August 1980).
8. G. I. Kerley, "Rational Function Method of Interpolation," Los Alamos Scientific Laboratory report LA-6903-MS (August 1977).

V. NEUTRONICS CALCULATIVE METHODS

In this chapter we describe the methods used to solve the equations for the SIMMER-II neutronics models. These methods are related to the techniques used in several other codes. The transport-theory methods closely follow those of TWOTRAN-IV,¹ and the diffusion-theory methods are similar to those of DIF3D,² as are some of the acceleration methods. Although the quasistatic algorithm is related to those of QX1³ and FX2,⁴ new features have been developed independently or extended to meet the specific requirements of transition-phase analysis. One unique feature that has been developed for the SIMMER-II neutronics model is the capability to start a calculation from a nonstationary reactor state. Thus, SIMMER-II can be used to continue accident calculations begun with multichannel accident analysis codes such as SAS3D^{5,6} without discontinuities in the reactivity and power at the transition time. Other features provide an improved capability to treat large transient flux-shape distortions and the long times often involved in transition-phase calculations.

A. The Spatial, Temporal, and Energy Mesh

The spatial mesh structure used for the neutronics calculations is based on the fluid-dynamics mesh, but the entire domain of the fluid-dynamics mesh need not be used. Thus, the computing expense can be reduced by eliminating regions where the neutronics effects are known to be negligible during the course of the transient. The regions to be considered are defined by specifying the fluid-dynamics cells that are the left, right, top, and bottom boundaries of the neutronics problem. For the neutronics methods, each mesh cell is considered to be homogeneous; that is, only averaged quantities are described.

For some problems, the size of the neutronics cells must be limited to obtain realistic flux shapes or to enhance iteration convergence. Thus, an option to subdivide specified rows or columns of fluid-dynamics cells is provided. The number of subdivisions desired for each row and column is user-specified, and the fluid-dynamics cells are subdivided into equal-volume neutronics cells.

The radial mesh is numbered from 1 to IM, with the mesh-cell centers indicated by subscript "i." Cell boundaries are indicated on the left by "i-1/2" and on the right by "i+1/2." The axial mesh is numbered from 1 to JM, with mesh centers indicated by subscript "j." Axial cell boundaries are indicated on the bottom by "j-1/2" and on the top by "j+1/2." For the discrete-ordinates transport-theory option, the angular mesh is divided into MM discrete intervals per quadrant, with boundaries indicated by the subscripts "m-1/2" and "m+1/2."

For the neutronics methods, the time variable is discretized to three levels of time steps to accommodate phenomena of differing rates. The largest time steps are associated with recalculation of spatial flux shapes, the intermediate steps are associated with an approximate update of the flux shape and the evaluation of integral parameters (for example, reactivity) describing the reactor state, and the smallest steps are associated with the reactor integral-power-level solution.

The neutronics and fluid-dynamics equations are partially decoupled; that is, the equations are not solved simultaneously at each time step. The neutronics state is updated at intervals determined by the fluid-dynamics solution for densities and temperatures at that time. Information transferred from the neutronics algorithm to the fluid-dynamics algorithm includes the cell-wise specific powers and the time dependence of the integral-power-level solution. The transfer from the fluid-dynamics algorithm to the neutronics algorithm consists of the densities and temperatures of the fluid-dynamics cell corresponding to each neutronics cell. The time-dependent behavior of the reactor power is extrapolated from current and previous neutronics solutions, modified to enforce energy conservation between the neutronics and fluid-dynamics solutions, and used for a series of fluid-dynamics steps. This approach permits the separation of the neutronics and fluid-dynamics methods and requires only the communication of such key quantities as component temperatures, densities, and powers.

In SIMMER-II, both the neutron-transport equation and the neutron-diffusion equation are solved as multigroup equations. The energy range is divided into IGM intervals of energy width ΔE_g , $g = 1, 2, \dots, IGM$, where $g = 1$ represents the highest energy group and $g = IGM$ represents the lowest energy group. The angular flux, ψ_g , and the scalar flux, ϕ_g , for group g are defined as

$$\psi_g = \int_{\Delta E_g} \psi(E) dE \quad , \text{ and} \quad (V-1)$$

$$\phi_g = \int_{\Delta E_g} \phi(E) dE \quad . \quad (V-2)$$

An average cross section for the group is defined as

$$\Sigma_{tg} = \int_{\Delta E_g} \Sigma_t(E) \Phi(E) \frac{dE}{\Phi_g} , \quad (V-3)$$

where the weighting spectrum $\Phi(E)$ normally is determined by a fine-group approximation such as a fundamental-mode calculation. This cross-section "collapse" is done with computer programs other than SIMMER-II. In SIMMER-II, the macroscopic cross section of Eq. (V-3) is calculated from microscopic cross sections by an isotopic self-shielding algorithm dependent on the time-dependent individual cell isotopic densities and temperatures computed by the fluid-dynamics methods. Note that for transport calculations, $\Phi(E)$ generally would have angular dependence. In this case, Σ_{tg} will depend on angle unless the angular dependence of $\Phi(E)$ is separable. However, no provision for such dependence is provided in SIMMER-II.

B. Methods

The methods involved in neutronics calculations include transient cross-section determination, solution of the time dependence of the neutron conservation equations, and solution of the space dependence. The cross sections represent the coupling from the fluid dynamics solution and vary as the cell-wise isotopic densities and temperatures vary. The time dependence of the neutron conservation equation is treated by the quasistatic technique in which the conservation equation is factored into two equations with differing degrees of time dependency. Two approximate forms of the neutron conservation equation are available in SIMMER-II: the S_N transport equation and the diffusion equation.

1. Cross-Section Methods. In SIMMER-II, several steps are involved in the cross-section calculations. Because of the time dependence of individual cell compositions and temperatures, microscopic cross sections are input as infinitely dilute, or unshielded, quantities. Cell-wise shielding factors are calculated according to the Bondarenko formalism⁷ after cell compositions and temperatures are known. Macroscopic cross sections for each mesh cell then are calculated by summing the various isotopic cross sections weighted by the appropriate isotopic concentrations and the previously determined shielding factors.

a. Neutronics/Fluid-Dynamics Coupling. The interaction between the fluid-dynamics and neutronics calculations occurs with information flowing in both directions. Changes in the material densities and temperatures affect the criticality of the reactor by changing the neutron cross sections, and the neutron flux affects material motion by time- and space-dependent energy deposition caused by neutron capture and fission processes.

The isotopic composition of each material component is fixed for each cross-section region. These composition data are used to calculate isotopic densities and temperatures as the material composition varies in time and space. The method is sufficiently general to allow the same isotope to exist in more than one material component. The total atom number density and average temperature for each isotope in a cell are determined by

$$\hat{N}_{ij}^i = \sum_{m=1}^{NC} N_{ij}^{im} \bar{\rho}_{ij}^m, \text{ and} \quad (V-4)$$

$$\bar{T}_{ij}^i = \frac{\sum_{m=1}^{NC} N_{ij}^{im} \bar{\rho}_{ij}^m \bar{T}_{ij}^m}{\hat{N}_{ij}^i}, \quad (V-5)$$

where

$\hat{N}_{ij}^i$ denotes the atom number density of isotope i for mesh cell ij ,

N_{ij}^{im} denotes the atom number density of isotope i in component m for each mesh cell ij (depends on ij because different cross-section regions, where component composition may vary, are allowed),

$\bar{\rho}_{ij}^m$ denotes the average density of component m for the mesh cell, and

$\bar{T}_{ij}^i$ denotes the average temperature of isotope i in each mesh cell ij , and

$\bar{T}_{ij}^m$ denotes the average temperature of component m in the mesh cell ij .

Changes in the resonance self-shielding of the cross sections are caused by changes in the temperature and/or the background from other isotopes. This is modeled by self-shielding factors that multiply the infinitely dilute cross sections to give the effective (self-shielded) microscopic cross sections. At each

reactivity time step, these effective microscopic cross sections are computed and mixed to form the various macroscopic cross sections required for each mesh cell. To determine the macroscopic cross section for a particular reaction in a mesh cell, the isotopic number densities and appropriate shielding factors multiply the infinitely dilute (unshielded) cross sections.

$$\Sigma_{x,ijg} = \sum_{i=1}^{LNISIP} \hat{N}_{ij}^i \sigma_{x,g}^i f_{x,ijg}^i \quad , \quad (V-6)$$

where

x denotes a particular reaction type,

$\sigma_{x,g}^i$ denotes the (input) infinitely dilute cross section of reaction type x for isotope i ,

$f_{x,ijg}^i$ denotes the shielding factor for isotope i for reaction type x evaluated for the background cross section and temperature of cell ij , and

$LNISIP$ denotes the total number of isotopes in the calculation.

The microscopic cross sections used in SIMMER-II are listed in Table XVII, and the shielding factors required for the cross sections are listed in Table XVIII.

The Bondarenko⁷ method uses a background cross section $\sigma_{0,ijg}^j$ to represent the total macroscopic cross section caused by all isotopes except for isotope j in a given mesh cell. The total macroscopic cross section for a group is

$$\Sigma_{t,ijg} = \hat{N}_{ij}^j \sigma_{t,ijg}^j + \sum_{k \neq j} \hat{N}_{ij}^k \sigma_{t,ijg}^k \quad , \quad (V-7)$$

where

$\sigma_{t,ijg}^i = f_{t,ijg}^i \sigma_{t,g}^i$ is the total microscopic shielded cross section for isotope i and

$f_{t,ijg}^i = f(\sigma_0, T)$ is the resonance self-shielding factor.

TABLE XVII

UNSHIELDED MICROSCOPIC CROSS SECTIONS USED BY SIMMER-II

<u>Symbol</u>	<u>Pointer</u>	<u>Definition</u>
σ_{tr}	LISTRN	P1 weighted transport (used only for diffusion theory)
σ_c	LISCAP	Capture (n, γ)
σ_f	LISFIS	Fission (n,f)
$\nu\sigma_f$	LISNFS	Neutron yield per fission * fission cross section
σ_{msc}	LISMSC	(n, α) + (n,p) + (n,D) + (n,T)
$\sigma_{e,g \rightarrow g}$	LISELI	In-group elastic scatter
$\sigma_{ne,g \rightarrow g}$	LISING	In-group inelastic scatter + in-group (n,2n) scatter
σ_r	LISREM	Summation of inelastic downscatters - ingroup (n,2n) scatter + summation of (n,2n) downscatter
$\sigma_{e,g \rightarrow g+1}$	LISELO	Elastic downscatter
$\sigma_{1e,g \rightarrow g}$	LISEI1	P1 component of in-group elastic scatter (present only for the P ₁ correction in transport theory)
$\sigma_{1e,g \rightarrow g+1}$	LISEO1	P1 component of elastic downscatter (present only for the P ₁ correction in transport theory)
$\sigma_{1ne,g \rightarrow g}$	LISII1	Summation of P1 components of in-group inelastic scatter and in-group (n,2n) scatter (present only for the P ₁ correction in transport theory)
σ_{1neo}	LISIO1	Summation over all lower energy groups of P1 components of inelastic downscatter and (n,2n) downscatter (present only for the P ₁ correction in transport theory)
$\sigma_{g' \rightarrow g}$	LISDWN+g-g'-1	Inelastic downscatter + (n,2n) downscatter
	LISMAX	Total number of microscopic cross-section types stored.

TABLE XVIII
ISOTOPIC SHIELDING FACTORS

<u>Symbol</u>	<u>Pointer</u>	<u>Definition</u>
f_f	LIFFF	Factor used to shield σ_f
f_c	LIFFC	Factor used to shield σ_c
f_e	LIFFE	Factor used to shield $\sigma_{e,g \rightarrow g}$ (and $\sigma_{e,g \rightarrow g+1}$ if NEUPREP is not defined)
f_d	LIFFD	Factor used to shield $\sigma_{e,g \rightarrow g+1}$ (present only if NEUPREP is defined)
f_{tr}	LIFFT	Factor used to shield σ_{tr} (present only for diffusion theory)
	LIFFB	Total number of shielding factor types stored (three or four if NEUPREP is not defined and four or five if NEUPREP is defined).

If Eq. (V-7) is divided by $\hat{N}_{ij}^j$,

$$\frac{\Sigma_{t,ijg}}{\hat{N}_{ij}^j} = \sigma_{t,ijg}^j + \frac{1}{\hat{N}_{ij}^j} \sum_{k \neq j} \hat{N}_{ij}^k \sigma_{t,ijg}^k \quad (V-8)$$

Then the second term on the right of Eq. (V-8) is defined as the background cross section $\sigma_{o,ijg}^j$; that is,

$$\sigma_{o,ijg}^j = \frac{1}{\hat{N}_{ij}^j} \sum_{k \neq j} \hat{N}_{ij}^k \sigma_{t,ijg}^k \quad (V-9)$$

The background cross section describes the effect of other isotopes in a mixture. The effect is especially significant in the vicinity of a large cross-section resonance. The background cross section cannot be determined until the total shielded cross sections of all isotopes are known. Because $\sigma_{x,ijg}^k$ is a

function of the shielding factor $f_{x,ijg}^k$ for isotope k and reaction type x and because $f_{x,ijg}^k$ depends on its background cross section, an iteration must be used to obtain both the shielding factors and the shielded total cross section. An initial guess for $\sigma_{0,ijg}^j$ is obtained by assuming the shielding factors to be approximated by unity,

$$\sigma_{0,ijg}^{1,j} = \frac{1}{\hat{N}_{ij}^j} \sum_{k \neq j} \hat{N}_{ij}^k (\sigma_{c,g}^k + \sigma_{e,g \rightarrow g+1}^k + \sigma_{r,g}^k + \sigma_{msc,g}^k + \sigma_{f,g}^k + \sigma_{ne,g \rightarrow g}^k + \sigma_{e,g \rightarrow g}^k) , \quad (V-10)$$

where the total cross section is replaced by its components and the nomenclature for the unshielded microscopic cross sections is given in Table XVII. Subsequent σ_0 estimates (iteration count n) are obtained from

$$\sigma_{0,ijg}^{n+1,j} = \frac{1}{\hat{N}_{ij}^j} \sum_{k \neq j} \hat{N}_{ij}^k (\sigma_{c,g}^k f_{c,ijg}^k + \sigma_{e,g \rightarrow g+1}^k f_{e,ijg}^k + \sigma_{r,g}^k + \sigma_{msc,g}^k + \sigma_{f,g}^k f_{f,ijg}^k + \sigma_{ne,g \rightarrow g}^k + \sigma_{e,g \rightarrow g}^k f_{e,ijg}^k) , \quad (V-11)$$

where the various shielding factors $f_{x,ijg}^k$ depend on the previous iteration for $\sigma_{0,ijg}^{n,k}$.

Equation (V-6) is not used in calculating the energy deposition cross sections. For neutron heating, the energy deposition is proportional to the mass of the component. During the fluid-dynamics calculation, the mass of each component in a cell may be changing continuously; hence, it is appropriate to calculate the energy source for each component as the energy per unit mass and then multiply this result by the component density during the fluid-dynamics calculation. The cross sections for determining energy deposition are calculated as

$$\Sigma_{e,ijg}^m = \sum_{i=1}^{LNISIP} N_{ij}^{im} (\alpha_c^i \sigma_{c,ijg}^i + \alpha_f^i \sigma_{f,ijg}^i) , \quad (V-12)$$

where

$\Sigma_{e,ijg}^m$ denotes energy deposition macroscopic cross section per unit density of component m for neutron energy group g in mesh cell (i,j),

α_c^i denotes the energy conversion factor per neutron capture in isotope i, and

α_f^i denotes the energy conversion factor per fission in isotope i.

After the neutron flux is calculated, the power per unit mass (W/kg) of component m in each cell is calculated as

$$Q_{Nmij} = \sum_{g=1}^{IGM} \Sigma_{e,ijg}^m \phi_{ijg} \quad , \quad (V-13)$$

where ϕ_{ijg} is the spatial flux shape. (The product of the flux shape and the flux amplitude equals the flux.) These power densities are passed to the fluid-dynamics calculation and are multiplied by the flux amplitude function and by the component densities in the structure-, liquid-, and vapor-field energy equations.

b. Use of the Neutronics Preprocessor. The cross-section coupling between the fluid-dynamics and neutronics calculations discussed earlier was described in terms of individual isotopes. The self-shielding of isotopic cross sections and the mixing of microscopic cross sections into macroscopic cross sections is presented in later sections. These operations can be performed with individual isotopes as the basic quantity. However, experience has shown that this approach can be very costly in terms of computer expense. In this section, an alternative approach is described that can result in significant improvement in both calculative cost and the physics treatment. The alternative approach is implemented by invoking the "NEUPREP" option (see Chap. VI) when creating the SIMMER-II source and by executing the MXS⁸ neutronics preprocessor before a SIMMER-II transient calculation. The isotopic approach used by previous versions of SIMMER-II is implemented by not invoking the "NEUPREP" option.

Several techniques⁹ are used in the NEUPREP approach. First, isotopic cross sections are premixed by the MXS preprocessor to create material cross sections. This permits the number of different cross sections to be reduced. Second, interpolation of the shielding factor background dependence has been replaced by the evaluation of a shielding "function." Third, quantities that need to be calculated only once for a SIMMER-II transient calculation have been placed in the MXS preprocessor. Fourth, the calculation of shielding factors and the mixing of macroscopic cross sections have been vectorized.

In this section, the concept of material cross sections is introduced, and the basic equations used are derived. The determination of material shielding factors is discussed in a later section. In this and the following sections, the notation "M" is used to indicate the material dependence of quantities related to material cross section. This "M" is different from the "m" used to annotate quantities associated with the fluid-dynamic components. A fluid-dynamics component (for example, fissile fuel) may be composed of several cross-section materials as well as isotopes.

(1) Introduction of the Material Cross-Section Method. The basic idea of using the cross-section mixing method to create material cross sections was first proposed by L. L. Smith in 1976.¹⁰ This idea is based on the fact that, in most reactor analyses, the isotopic composition of components such as fertile and fissile fuel or steel is constant over the time period of a reactor accident. In SIMMER-II, several such components can be defined by the user, and the isotopic composition of each component is held constant; however, the amount of different components in a given spatial cell may change continuously. Thus, the calculation of the effective macroscopic cross section in SIMMER-II can be changed from an isotopic approach to an approach based on material cross sections: in the SIMMER-II setup, material quantities are simply substituted for isotopic quantities. The results will be the same for both approaches (except for the effects of some minor approximations) unless two or more materials contain the same isotopes. This problem (which is termed "the common-isotope problem") is discussed in Appendix R.

Macroscopic cross sections should agree for both approaches. A particular macroscopic cross section is calculated in the isotopic approach by

$$\Sigma_{x,ijg} = \sum_{i=1}^{\text{Number of isotopes}} \hat{N}_{ij}^i \sigma_{x,ijg}^i .$$

The same quantity in the material approach is given by

$$\Sigma_{x,ijg} = \sum_{M=1}^{\text{Number of materials}} \hat{N}_{ij}^M \sigma_{x,ijg}^M .$$

where

$\hat{N}_{ij}^M$ denotes the number density of material M in mesh cell ij and
 $\sigma_{x,ijg}^M$ denotes the microscopic shielded cross section for isotope M in cell ij.

Because the number of isotopes usually used in LMFBR accident analysis typically can be about 15 to 20 and the number of materials used is about 4 to 5, the efficiency of effective cross section calculation could potentially be improved by a factor of 3 to 4 by this technique alone.

(2) Basic Equations of Cross-Section Mixing. The mixing equations are derived by making the reaction rates based on the material calculation agree with those based on an isotopic calculation.

Infinite Cross Sections. The number density of a material is defined as

$$\hat{N}_{ij}^M = \sum_{i \in M} \hat{N}_{ij}^i \quad . \quad (V-14)$$

Using the material number density, conservation of reaction rates is expressed as

$$\begin{aligned} (\text{Reaction Rate } x \text{ of Material } M) &= \sum_{i \in M} \hat{N}_{ij}^i \sigma_{x,g}^i \phi_{ijg} \\ &= \hat{N}_{ij}^M \sigma_{x,g}^M \phi_{ijg} \quad . \end{aligned}$$

Therefore, for material M,

$$\sigma_{x,g}^M = \frac{\sum_{i \in M} \hat{N}_{ij}^i \sigma_{x,g}^i}{\hat{N}_{ij}^M} \quad . \quad (V-15)$$

Shielding Factors. An isotopic background cross section is related to a material background cross section as follows. Equation (V-9) gives the $\sigma_{0,ijg}^j$ calculated from isotopic quantities. In terms of material quantities, $\sigma_{0,ijg}^j$ is given by

$$\sigma_{0,ijg}^j = \frac{1}{\hat{N}_{ij}^j} \sum_{k \neq j} \hat{N}_{ij}^k \sigma_{t,ijg}^k$$

$$= \frac{1}{\hat{N}_{ij}^j} \sum_{\substack{k \neq j \\ k \in M}} \hat{N}_{ij}^k \sigma_{t,ijg}^k + \frac{\hat{N}_{ij}^M}{\hat{N}_{ij}^j} \sigma_{0,ijg}^M \quad (V-16)$$

where $\sigma_{0,ijg}^M$ is the material background cross section of material "M" and is defined as

$$\sigma_{0,ijg}^M = \frac{1}{\hat{N}_{ij}^M} \sum_{M' \neq M} \hat{N}_{ij}^{M'} \sigma_{t,ijg}^{M'} \quad (V-17)$$

Equation (V-16) can be solved for $\sigma_{0,ijg}^j$ when $\sigma_{0,ijg}^M$ is given because $\sigma_{t,ijg}^k$ is a function of $\sigma_{0,ijg}^j$. The solution method used for Eq. (V-16) is an iterative calculation because of its nonlinearity.

Using the solution of $\sigma_{0,ijg}^j$ from Eq. (V-16), the conservation of reaction rates for a particular reaction, x, is expressed as

$$\begin{aligned} (\text{Reaction Rate } x \text{ of Material } M) &= \sum_{i \in M} \hat{N}_{ij}^i f_{x,ijg}^i (\sigma_{0,ijg}^i, \bar{T}_{ij}^i) \sigma_{x,g}^i \phi_{ijg} \\ &= \hat{N}_{ij}^M F_{x,ijg}^M (\sigma_{0,ijg}^M, \bar{T}_{ij}^M) \sigma_{x,g}^M \phi_{ijg} \end{aligned}$$

where $F_{x,ijg}^M$ denotes the self-shielding factor for material M in cell ij; therefore,

$$F_{x,ijg}^M (\sigma_{0,ijg}^M, \bar{T}_{ij}^M) = \frac{\sum_{i \in M} \hat{N}_{ij}^i f_{x,ijg}^i (\sigma_{0,ijg}^j, \bar{T}_{ij}^i) \sigma_{x,g}^i}{\hat{N}_{ij}^M \sigma_{x,g}^M} \quad (V-18)$$

(Fission neutron source resulting from material M) =

$$\sum_{i \in M} x_{pg}^i \sum_{g'=1}^{IGM} \hat{N}_{ij}^i v_{g'}^i f_{f,ijg'}^i \sigma_{f,g'}^i \phi_{ijg'}$$

(V-20)

$$= x_{pg}^M \sum_{g'=1}^{IGM} \hat{N}_{ij}^M v_{g'}^M F_{f,ijg'}^M \sigma_{f,g'}^M \phi_{ijg'}$$

Equation (V-20) implies that the exact x_{pg}^M must be a function of flux spectrum ϕ_{ijg} as well as $\sigma_{o,ijg}^M$. However, the isotopic dependency of x_g^i is always very small, so the x_{pg} of the most important fissile isotope for all fissionable materials can be used. This approach is the same as that used elsewhere in SIMMER-II for x and is sufficiently accurate.

Energy Deposition. The energy deposition factors, α_f and α_c , are defined as the generated thermal energy per fission reaction or capture reaction, respectively. Cross sections for calculating energy deposition are defined by Eq. (V-12).

The conservation equation for energy deposition used to define the material energy deposition factors is,

$$(\text{Generated power per unit volume}) = \sum_{i \in M} \alpha_x^i \hat{N}_{ij}^i f_{x,ijg}^i \sigma_{x,g}^i \phi_{ijg}$$

(V-21)

$$= \alpha_{x,g}^M \hat{N}_{ij}^M F_{x,ijg}^M \sigma_{x,g}^M \phi_{ijg}$$

where $x = f$ (fission) or c (capture).

It is only a small approximation (in an integral sense) to eliminate the shielding factor in Eq. (V-21), thus allowing the α 's to apply to all cells. Solving for α ,

$$\alpha_{x,g}^M = \frac{\sum_{i \in M} \alpha_x^i \hat{N}_{ij}^i \sigma_{x,g}^i}{\hat{N}_{ij}^M \sigma_{x,g}^M}$$

(V-22)

(Fission neutron source resulting from material M) =

$$\begin{aligned} & \sum_{i \in M} x_{pg}^i \sum_{g'=1}^{IGM} \hat{N}_{ij}^i v_{g'}^i f_{f,ijg}^i \sigma_{f,g'}^i \phi_{ijg} \\ & = x_{pg}^M \sum_{g'=1}^{IGM} \hat{N}_{ij}^M v_{g'}^M F_{f,ijg}^M \sigma_{f,g'}^M \phi_{ijg} \end{aligned} \quad (V-20)$$

Equation (V-20) implies that the exact x_{pg}^M must be a function of flux spectrum ϕ_{ijg} as well as $\sigma_{o,ijg}^M$. However, the isotopic dependency of x_g^i is always very small, so the x_{pg} of the most important fissile isotope for all fissionable materials can be used. This approach is the same as that used elsewhere in SIMMER-II for x and is sufficiently accurate.

Energy Deposition. The energy deposition factors, α_f and α_c , are defined as the generated thermal energy per fission reaction or capture reaction, respectively. Cross sections for calculating energy deposition are defined by Eq. (V-12).

The conservation equation for energy deposition used to define the material energy deposition factors is,

$$\begin{aligned} (\text{Generated power per unit volume}) &= \sum_{i \in M} \alpha_x^i \hat{N}_{ij}^i f_{x,ijg}^i \sigma_{x,g}^i \phi_{ijg} \\ &= \alpha_{x,g}^M \hat{N}_{ij}^M F_{x,ijg}^M \sigma_{x,g}^M \phi_{ijg} \end{aligned} \quad (V-21)$$

where $x = f$ (fission) or c (capture).

It is only a small approximation (in an integral sense) to eliminate the shielding factor in Eq. (V-21), thus allowing the α 's to apply to all cells. Solving for α ,

$$\alpha_{x,g}^M \approx \frac{\sum_{i \in M} \alpha_x^i \hat{N}_{ij}^i \sigma_{x,g}^i}{\hat{N}_{ij}^M \sigma_{x,g}^M} \quad (V-22)$$

Note that in Eq. (V-22) the material α is a function of energy group. Only a trivial generalization of Eq. (V-12) is required to treat energy-dependent α 's.

c. Determination of Shielding Factors (NEUPREP-Off Method). The shielding factors are obtained by interpolation based on the values of $\sigma_{0,ijg}^{n,j}$ from the previous iteration as defined by Eq. (V-11). Convergence of this iteration procedure is normally rapid, usually requiring only two or three iterations. The iteration is continued until the largest fractional change in any $\sigma_{0,ijg}^{n+1,j}$ between iterations is less than an input maximum, EPSBKG. The maximum number of iterations is limited by a second input quantity, ITLMBG.

Both the background and temperature interpolations are Lagrangian. A variable-order algorithm is used with the number of points entered as input values: ISIGOD for background interpolation and ITEMOD for the temperature interpolation. No extrapolation is performed if the isotopic background cross section or temperature falls outside the range of shielding factors provided. In this case, the end table-point value is used.

The Lagrangian method was chosen because it is simple, thus saving calculative effort, and because it provides sufficient accuracy. However, both interpolations are identified and isolated in the code for users who may need to develop a more sophisticated scheme.

d. Determination of Shielding Factors (NEUPREP-On Method). When the NEUPREP option is used, the preprocessor mixes isotopic quantities into material quantities. The difference between isotopic cross sections and material cross sections is invisible to SIMMER-II; Eqs. (7-9) through (V-11), are used for the background cross section and the material shielding factors are interpolated without distinction. Isotopes and materials can be substituted or mixed without affecting the results, except for the possibility of common isotopes. Techniques for dealing with this problem are discussed in Appendix R.

Although the interpolation process used with the NEUPREP option has the same function, input, and output as that described in the previous section, the techniques used are entirely different. These techniques are much more complicated than the simple Lagrangian method described in the previous section but result in significant efficiency gains and an improved physics treatment.

(1) Approach for background interpolation. The method used in SIMMER-II is based on the application of several techniques to improve both efficiency and the physics treatment.

Least-Squares Fitting. Various studies (see for example Ref. 12) have proposed many different kinds of background interpolation schemes, including physics-based and empirical (mathematics-based) formulas. The common idea (which was postulated implicitly in these schemes) is to trace or intersect each table point exactly. However, after studying actual shielding-factor data, it was not considered necessary to intersect each table point exactly; it is more important to assure a certain accuracy over the full range of σ_0 from zero to infinity. A least-square fitting accomplishes this and is used to determine the best formula for a set of shielding factor data tables. Because the least-square fitting need be performed only once for a SIMMER-II calculation, it is located in the MXS preprocessor.

A Unique Function for the Domain of σ_0 . In other studies, several table points usually were selected from the vicinity of a needed σ_0 value to interpolate the shielding factors. However, when the efficiency of programming using vectorization techniques is considered, this discretization of the σ_0 -domain is an obstacle. Therefore, a unique function for the entire domain of σ_0 for a set of shielding factors is used whenever possible.

Multifunction Methods. Previous versions of SIMMER-II used "mathematics-based" interpolation functions (that is, Lagrange formulas). The physical information content of interpolation formulas, such as Ott's,¹³ can be used to reduce the calculative effort significantly. However, extensive numerical experiments⁹ have been performed showing that no single such formula can be used to treat all situations that may be encountered. This is mainly because such formulas represent approximations of the complex physics involved in nuclear resonance theory. It was determined that a collection of formulas could treat a majority of cases and that the remainder could be treated with mathematics-based formulas, for example, simple polynomials or, as a last resort, B-splines. A decision theory was developed to assign the "best" formula to each different shielding factor data table.

Precalculation of Coefficients. Many of the steps involved in generating the special interpolation functions, for example, determining coefficients for the background dependence, need be performed only once for each cross-section set or possibly only once for a SIMMER-II transient calculation. A detailed decision theory (described in Ref. 8) is used to select the proper interpolation function for each shielding-factor data table. The MXS preprocessor supplies SIMMER-II with the coefficient table of each shielding factor and with a formula flag to indicate which function to use for a given evaluation. The shielding-

factor data table itself is not used in SIMMER-II. Then, during transient calculations in SIMMER-II, the best formula is selected inexpensively by the preprocessor-determined flag.

(2) Approach for Temperature Interpolation. The problem of temperature interpolation has several features different from the problem of background interpolation. The following are the most important differences.

1. Temperature interpolation is performed inside SIMMER-II after background interpolation; therefore, the coefficients cannot be precalculated by the preprocessor, and the efficiency of finding the coefficients becomes more important relative to accuracy.
2. Fortunately, the temperature dependency of the shielding factor table is generally much simpler than the background dependency.
3. The temperature dependency determines the Doppler reactivity ($\frac{1}{T} \frac{dk_0}{dT}$), so the first derivative ($\frac{df}{dT}$) is as important as the shielding factor value itself.
4. Unlike background interpolation, there seems to be no strong basis for any particular "physics-based" formula for temperature interpolation in the literature.
5. Typical nuclear data sets have rather low temperatures as the maximum point, for example, ABBN = 2100 K and JFS-3 = 4500 K. The only exception is the Los Alamos LIB-IV-based data, which has 8000 K data as the maximum. Therefore, some method of temperature extrapolation is needed, unlike for background interpolation.

As a result of these features, the following approach was used. One of several simple (mathematics-based only) functions is used to perform the temperature interpolation from the results of the background interpolations. A decision theory has been developed and implemented in the MXS preprocessor to predict which formula is the best for a given set of shielding factor data. A formula flag table is transmitted to SIMMER-II so that the best formula is selected during transient calculations without repeating the expensive decision-making process. An extrapolation treatment also has been developed to extend the temperature dependence of the shielding factors.

(3) Background Interpolation of Shielding Factors.

Formula Selection for the σ_0 -Interpolation. For the σ_0 -interpolation, 15 different formulas (chosen in part from information contained in Refs. 12-18) are used in SIMMER-II. Table XIX shows these and their assigned priorities. In particular, the following comments are applicable.

TABLE XIX
INTERPOLATION FORMULAS CHOSEN FOR THE
BACKGROUND CROSS SECTION

<u>Classification</u>	<u>Name</u>	<u>Formula</u>	<u>Priority</u>	<u>Number of Coefficients</u>
Physics- Based Formulas	Constant	$F = A$	1	1
	Linear	$F = AX + B$	2	2
	K. Ott	$F = \left(\frac{\sigma_0 + B}{\sigma_0 + A} \right)^{1/2}$	3	2
	3-point Segev	$F = C + \left(\frac{\sigma_0 + B}{\sigma_0 + A} \right)^{1/2}$	4	3
	4-point Segev	$F = C + D \left(\frac{\sigma_0 + B}{\sigma_0 + A} \right)^{1/2}$	5	4
Mathematics- Based Formulas	Hyperbolic Tangent	$F = A \tanh [B(X + C)] + D$	6	4
	Rational	$F = \frac{CX^2 + DX + E}{AX^2 + BX + 1}$	7	5
	2nd Polynomial	$F = \sum_{i=1}^3 A_i X^{i-1}$	8	3
	3rd Polynomial	$F = \sum_{i=1}^4 A_i X^{i-1}$	9	4
	4th Polynomial	$F = \sum_{i=1}^5 A_i X^{i-1}$	10	5
	5th Polynomial	$F = \sum_{i=1}^6 A_i X^{i-1}$	11	6

TABLE XIX (CONT)

INTERPOLATION FORMULAS CHOSEN FOR THE
BACKGROUND CROSS SECTION

<u>Classification</u>	<u>Name</u>	<u>Formula</u>	<u>Priority</u>	<u>Number of Coefficients</u>
	6th Polynomial	$F = \sum_{i=1}^7 A_i X^{i-1}$	12	7
	7th Polynomial	$F = \sum_{i=1}^8 A_i X^{i-1}$	13	8
	8th Polynomial	$F = \sum_{i=1}^9 A_i X^{i-1}$	14	9
	Cubic B-Spline	$F = \sum_{i=1}^{N+2} A_i B_i(X)$	15	N+2

where

X denotes $\log_{10} \sigma_0$.

N denotes number of σ_0 -table points.

$B_i(X)$ denotes basis functions of the B-spline interpolation function, and

A, B, C, D, E , and A_i denote coefficients.

- Higher priorities are assigned to the physics-based formulas than to the mathematics-based formulas because of their inherent potential accuracy and desired characteristics. Because of the assumptions (an isolated resonance, the natural line shape, and the validity of the narrow resonance approximation) used for the derivation of the

physics-based formulas, these formulas are not valid for all shielding-factors.

- The polynomial functions from the second through the eighth degree were included because of their computational efficiency, but their ability to fit the shielding-factors is limited.
- All the formulas except the B-spline¹⁹ are fitted using a least-square method. Only the B-spline is an "exact-data-tracing" method, and it was assigned the lowest priority because of its high computational cost. (However, the cost is not prohibitive because an efficient vectorization technique was developed.)

Method of Formula Selection for the σ_0 -Interpolation. The method used in the MXS preprocessor to select the "best" formula for each shielding factor is an elimination procedure. First, the shielding-factor table is fitted to the formula with the highest priority. If the results satisfy the success criteria, this formula is selected to represent the shielding factor data. If this formula fails, the shielding-factor table data are fitted to the formula with the next highest priority. If the results satisfy the success criteria, then that formula is selected to represent the shielding factor data. If that formula fails, the procedure is continued using the formula with the next highest priority until a "best" formula is found. A formula is successful if the maximum relative error of the fitted data compared with the reference data is less than the maximum permitted error ϵ_0 (an MXS user-input quantity):

$$\text{Max } \left| \frac{f_{\text{fitted}} - f_{\text{reference}}}{f_{\text{reference}}} \right| < \epsilon_0 \quad . \quad (\text{V-23})$$

The reference value used to evaluate the success criteria for the shielding functions is selected differently for the physics-based formulas than for the mathematics-based ones. For the physics-based formulas, only the shielding-factor-table points are used as reference values. For the mathematics-based formulas, reference points are interpolated by the B-spline function using the shielding-factor table points, yielding 10 intermediate points between each shielding-factor table point.

The permitted error, ϵ_0 , is defined as function of temperature so that the user can select different values of ϵ_0 for different temperatures according to the objective of his analysis.

(4) Temperature Interpolation of Shielding Factors.

Formula Selection for the Temperature Interpolation. Four different formulas are used in SIMMER-II for the temperature interpolation. Table XX shows these formulas and their priorities, which were assigned based on efficiency considerations.

- All of these formulas are mathematics-based. Unlike the σ_0 treatment, there are no good physics-based formulas that can be used for temperature-interpolation.

TABLE XX

TEMPERATURE INTERPOLATION FORMULAS

<u>Classification</u>	<u>Name</u>	<u>Formula</u>	<u>Priority</u>	<u>Number of Coefficients</u>
Mathematics-Based Formulas	Constant	$F = A$	1	1
	Linear	$F = A (\ln T) + B$	2	2
	Parabola	$F = A (\ln T)^2 + B (\ln T) + C$	3	3
	Cubic B-spline	$F = \sum_{i=1}^{N+2} A_i B_i (\ln T)$	4	N+2

where

N denotes number of temperature table points,

$B_i (\ln T)$ denote basis functions of the B-spline interpolation function, and

A, B, C, and A_i denote coefficients.

- The first through third formulas can cover more than 90% of the shielding-factor-table data with sufficient accuracy, as determined by many numerical tests⁹ of actual nuclear data. The largest error in the remaining cases always occurs in the lowest temperature region. For purposes typical of SIMMER-II calculations, normally such error is not important. In most cases, it is completely negligible; however, for unusual purposes, the B-spline function is included because it can fit any kind of data. Use of the B-spline is a user option because it has a very high calculative cost. (No vectorization technique has been developed because the B-spline coefficients cannot be precalculated in the MXS preprocessor.)
- The MXS preprocessor supplies a temperature-flag table that indicates the best formula for the temperature interpolation inside SIMMER-II.

Extrapolation Treatment. The preprocessor extrapolates shielding-factor-table data up to a user-supplied maximum temperature. This extrapolation uses the first derivative at the data point for the maximum temperature of the original isotopic data set, which is obtained by the B-spline fitting. The temperature where the preprocessor begins the extrapolation of the shielding-factor tables is transferred to SIMMER-II as the "switch temperature." SIMMER-II interpolates shielding factors above the switch temperature using only a log-linear function regardless of their temperature flags.

Method of Formula Selection for the Temperature Interpolation. The method used to select the best formula from among the four is the same as the "elimination procedure" used for the σ_0 interpolation. The success of a formula is determined by comparing the relative error of the interpolation function with ϵ_1 (Eq. (V-24)) and the absolute error of the first derivative of the interpolation function with ϵ_2 (Eq. (V-25)).

$$\max \frac{f_{\text{interpolation}} - f_{\text{reference}}}{f_{\text{reference}}} < \epsilon_1 \quad . \quad (\text{V-24})$$

$$\max \frac{\partial f_{\text{interpolation}}}{\partial (\ln T)} - \frac{\partial f_{\text{reference}}}{\partial (\ln T)} < \epsilon_2 \quad . \quad (\text{V-25})$$

The comparison is performed at intermediate points and at the table points. Only when a formula satisfies both Eqs. (V-24) and (V-25) is it chosen as an interpolation formula. The values interpolated by the B-spline functions are used as a reference.

e. Calculation of Macroscopic Cross Sections. After the shielding factors have been determined from the compositions and temperatures for a particular cell, the macroscopic cross sections are calculated. This step involves summing the product of the unshielded cross sections, the shielding factors, and the isotopic densities. The formulas required for the different macroscopic cross sections are given in Table XXI.

f. Calculative Approach Used by SHLDXS (NEUPREP-Off Method). Shielding factors are interpolated first over background and then over temperature. The interpolation is logarithmic for background and linear for temperature. First, the range of tabular values is searched to determine the tabulation points closest to the isotopic temperature and background values desired. Next, the background interpolated values are obtained for each tabular temperature point. Finally, a second interpolation over temperature is performed. A detailed flow chart of this process is provided in Figs. 15(a) and 15(b).

g. Calculative Approach Used by SHLDXS (NEUPREP-On Method). The overall sequence of calculations is initialization, determination of cell isotopic densities and temperatures, construction of a map of cells by temperature for each isotope, precalculation of temperature interpolation coefficients, determination of background cross sections, interpolation of background shielding factors, interpolation of temperature shielding factors, and finally calculation of shielded macroscopic cross sections. Within the sequence there are loops to process each energy group and to converge the background cross sections. The processing of all isotopes and reaction types also is performed by simple looping. Figure 16 shows the overall structure and the simple, outer loops.

To obtain vectorizable inner loops without resorting to extensive GATHER/SCATTER schemes, the cells are ordered according to temperature for each isotope. Because the background interpolation functions have coefficients that depend only on energy group, isotope, reaction type, and temperature, loops over all cells within a given isotope within a given temperature range can be vectorized. (Interpolation to the required temperature is performed later.) Because the formula type depends only on energy group, isotope, and reaction type, the formula flag evaluation and branch to the correct formula coding occurs within the reaction-type loop and before the temperature loop. (The loop

TABLE XXI
MACROSCOPIC CROSS SECTIONS

<u>Symbol</u>	<u>Pointer</u>	<u>Definition*</u>
$\Sigma_{s,g \rightarrow g}$	LMTIGS	$\sum_{i=1}^{LNISIP} \hat{N}^i (\sigma_{ne,g \rightarrow g}^i + \sigma_{e,g \rightarrow g}^i f_{e,g}^i) \quad (\text{Used only for transport theory.})$ <p>If the P_1 approximation is used, the term</p>
		$\sum_{i=1}^{LNISIP} \hat{N}^i (\sigma_{1e,g \rightarrow g}^i + \sigma_{1e,g \rightarrow g+1}^i) f_{e,g}^i \text{ is included.}$
$\Sigma_{s,g' \rightarrow g}$	LMTDWN+g-g'+1	$\sum_{i=1}^{LNISIP} \hat{N}^i \sigma_{g' \rightarrow g}^i$ <p>If $g=g'+1$, the term</p>
		$\sum_{i=1}^{LNISIP} \hat{N}^i \sigma_{e,g' \rightarrow g'+1}^i f_{e,g'}^i \text{ is included.}$
$\Sigma_{e,g}^m$	LMTENE+N-1	$\sum_{i=1}^{LNISIP} N^{im} (\alpha_c^i \sigma_c^i + \alpha_f^i \sigma_f^i)$
$\Sigma_{tr,g}$	LMTTRN	$\sum_{i=1}^{LNISIP} \hat{N}^i \sigma_{tr,g}^i f_{r,g}^i \quad (\text{not calculated if diffusion theory is not used.})$
$(\nu_p \Sigma_f)_g$	LMINSF	$\sum_{i=1}^{LNISIP} \hat{N}^i (\nu_p^i \sigma_f^i) f_{f,g}^i$
$\Sigma_{r,g}$	LMTREM	$\sum_{i=1}^{LNISIP} \hat{N}^i (\sigma_{c,g}^i f_{c,g}^i + \sigma_{e,g \rightarrow g+1}^i f_{e,g}^i + \sigma_{r,g}^i + \sigma_{msc,g}^i + \sigma_{f,g}^i f_{f,g}^i + \sigma_{ne,g \rightarrow g}^i + \sigma_{e,g \rightarrow g}^i f_{e,g}^i) \quad (\text{calculated only for the diffusion theory}).$
Σ_t	LMTOT	$\Sigma_{r,g} + \Sigma_{s,g \rightarrow g} \quad (\text{calculated only for transport theory})$

TABLE XXI (CONT)

MACROSCOPIC CROSS SECTIONS

*The cell-wise index (ij) has been dropped and the following apply.

- (1) The i index refers to input isotopes.
- (2) The m index refers to components.
- (3) $\hat{N}^i$ is the cell-averaged number density for isotope i,
- (4) N^{im} is the number density for isotope i in component m,
- (5) α_c^i is the thermal yield for neutron capture for isotope i,
- (6) α_f^i is the thermal yield for fission of isotope i.

over the number of temperatures is needed for later interpolation over temperature.) This structure is shown in Fig. 17.

The results of the background interpolations are used to perform the temperature interpolation. Still within the same group, isotope, and reaction type loops used for the background interpolations, the flag to identify the correct temperature interpolation formula is evaluated, and a branch to the appropriate coding is taken. Vector loops over all cells with a given isotope within a given temperature range again are possible because the results of the background interpolation have been stored in the correct order and the temperature interpolation coefficients have been precomputed for all isotopes in all cells. After temperature interpolation, the final shielding factors are SCATTERed into the non-temperature-ordered storage required for the subsequent calculation of macroscopic cross sections. The temperature interpolation algorithm is shown in Fig. 18.

Still within the energy group loop, the macroscopic cross sections are calculated (and the finite-difference coefficients if the diffusion option is defined). Principal cross sections are determined first, then down-scatter and component power cross sections. For each type of cross section, there is an outer loop over isotopes and an inner vectorized loop over mesh cells. No attempt has been made to skip over cells with a zero density for a given isotope because this would inhibit vectorization, and the vector gain is, on the average, greater than the potential savings of skipping empty cells.

Fig. 15(a).
Overall structure of the cross-section
shielding and mixing calculations (NEUPREP-off option).

Fig. 15(b).

Structure of the background and temperature interpolation portion of the cross-section shielding-factor calculation (NEUPREP-off option).

Fig. 16.
Structure of cross-section shielding
and mixing calculations (NEUPREP-on option).

Fig. 17.
Structure of the background interpolation portion
of the shielding-factor calculations (NEUPREP-on option).

Fig. 18.
Structure of the temperature interpolation portion
of the shielding-factor calculations (NEUPREP-on option).

The remainder of the subroutine is used for special optional edits, including shielding calculation performance statistics, a detailed print of isotopic cross sections and shielding factors for specified cells, a print of cell-wise macroscopic cross sections, and calculation and printing of regional isotopic inventories. This coding is located at the end of the subroutine to improve readability and execution efficiency for cases when editing is not desired.

2. Quasistatic Method. The quasistatic method²⁰ is used in SIMMEK-II to treat the time dependence of both the neutron diffusion and the discrete-ordinates transport equations. An alternative approach would be to finite-difference the time dependence of these equations in much the same way as the space dependence has been finite-differenced. However, very short time steps are required because of the mathematical "stiffness" of the resulting equations. This disadvantage is circumvented in the quasistatic approach; the flux is factored into a flux-shape equation and an amplitude equation. The approach is based on the fact that the power level, or flux amplitude, changes much more rapidly than the power distribution, or the flux spatial shape. Thus, the amplitude equation contains most of time dependence, where it can be treated easily because it is an ordinary differential equation. The flux-shape equation requires most of the computational effort; however, it is solved only as needed, that is, when transient flux shape change becomes significant.

a. General Approach. Because the quasistatic method treats the flux time dependence (which is caused by fluid-dynamic changes and causes future fluid-dynamics changes), it must be coupled closely to the fluid-dynamics models. Although a simultaneous solution of fluid dynamic and neutronics equations formally is required, in practice it is impossible with current computer resources. However, a reasonably accurate approximate treatment has been developed for SIMMER-II. The fluid dynamics solution is based on the results of flux amplitude calculations with extrapolated neutronics parameters. This solution is interrupted periodically to update the neutronics parameters based on the changes in material distributions and temperatures that have occurred. Because flux-shape and amplitude changes occur with different characteristic time scales, the quasistatic algorithm is well suited to respond to varying degree of fluid dynamic-neutronics coupling.

The quasistatic algorithm is based on a three-tiered time-step structure: the smallest time steps apply only to the amplitude-equation solution, the intermediate steps (reactivity steps) are associated with evaluating the shape

equation and amplitude equation coupling terms, and the largest steps (shape steps) are associated with the flux-shape solution interval. Within this time-step structure, the quasistatic algorithm controls the sequence of tasks required. For example, solution of the amplitude equation is required to extrapolate the power in preparation for the next series of fluid-dynamic steps, to update coupling terms after the fluid dynamics step, and to converge the iteration required at a flux-shape calculation step. This procedure is outlined in this section and explained in detail in a later section.

(1) Flux Factorization. The approach used in the quasistatic method is based on factoring the time-dependent spatial flux into a scalar amplitude function that contains the major time dependence and a spatial flux shape that contains all the space dependence of the flux but is slowly varying. The time-dependent flux shape may be either the angular flux, Ψ , or the total flux Φ ; the factorization approach is the same in both cases. Thus, the flux is

$$\Psi(t, r, z, \vec{\Omega}, E) = N(t) \psi(t, r, z, \vec{\Omega}, E), \quad \text{or} \quad (V-26)$$

$$\Phi(t, r, z, E) = N(t) \phi(t, r, z, E), \quad (V-27)$$

where N is the amplitude function and $N(t = 0) = 1$. The use of the flux-shape constraint condition

$$\sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \frac{\phi_{ijg}^* \phi_{ijg}}{v_g} = 1, \quad (V28)$$

where

v_g is the average neutron velocity in energy group g ,

v_{ij} is the volume of mesh cell (i,j) , and
 ϕ_{ijg}^* is the integrated adjoint neutron flux-shape function for cell (i,j) and energy group g ,

makes the factorization unique.

(2) Fluid-Dynamics/Neutronics Coupling. The neutronics and fluid dynamics solutions are coupled by a time-dependent amplitude function and spatial distribution of specific material powers for use during a series of fluid-dynamics steps; in turn, these neutronics parameters depend on the current fluid-dynamic configuration. Prediction of the amplitude-function time dependence is based on an extrapolation of the previous neutronics history. The validity of this approach requires that changes in the power level result from phenomena that do not change in character instantaneously. For example, fluid motions have an associated momentum that changes as forces act over some period of time. Likewise, loss of reactivity because of the Doppler effect requires energy integration over a period of time. However, step changes are possible, and the algorithm does provide for their treatment. The spatial distribution of specific material powers is assumed to change slowly because it depends on the flux shape. Thus, although its magnitude changes, this distribution remains unchanged during a series of fluid-dynamic steps.

After a series of fluid-dynamic steps, the reactivity is recalculated based on the new material densities and temperatures predicted by the fluid-dynamic calculation. The amplitude function is recalculated and compared with the previously predicted function actually used in the fluid-dynamics calculations. Formally correct treatment of the neutronics/fluid-dynamic coupling would require an iteration to converge these two quantities. However, recalculation of the series of fluid-dynamic steps would be too expensive. If the difference between the predicted and updated amplitude function is not great, only a small error is made by adjusting the amplitude-function projection for the next series of fluid-dynamic steps to account for the energy mismatch. This approach limits the number of fluid-dynamic steps that can be used with a single amplitude-function projection.

After a number of reactivity steps, the quasistatic constraint integral may begin to deviate significantly from unity. This is a result of the flux shape changing in response to the material motion and temperature changes; the neutron sources and sinks in the amplitude equation become out of balance with the sources and sinks in the shape equation. Therefore, the flux should be refactored. This is accomplished by performing a 7-iteration at the end of a shape step, where the amplitude function used over the shape step is modified by a series of flux-shape recalculations until the quasistatic constraint integral is satisfied. The modification of the amplitude function again introduces a discrepancy between the energy integrated in the fluid-dynamics calculation and

that predicted in the neutronics calculation because the fluid-dynamics calculation is not included in the iteration. As in the reactivity-step approach, this discrepancy is treated by adjusting the amplitude function used in the next series of fluid-dynamic steps rather than performing an iteration between the fluid dynamics and neutronics. The amplitude function is projected for a series of fluid-dynamic steps by first extrapolating the reactivity state and then solving the amplitude equations. The reactivity state time dependence is modeled by a parabola with coefficients determined by the reactivity at two previous reactivity steps. After the amplitude equation is solved, a parabola in the logarithm of the amplitude is fitted to the extrapolated point and two previous ones. This function then is evaluated in the fluid-dynamics calculation as needed.

The spatial distribution of material-specific powers is based on the flux-shape and cell cross sections at the most recent reactivity step. Cell power shape is calculated in the fluid-dynamics routines by multiplying the material-specific powers by their respective masses and summing over materials. The actual power then is obtained by multiplication with the amplitude function. This results in a first-order-accurate prediction of the effects of fluid motion and temperature change, with only the corresponding flux-shape changes and cross-section shielding changes neglected during the fluid-dynamics calculation.

Cell-dependent cross-section determination for reactivity calculation or flux-shape updates requires the cell-dependent isotopic composition and temperatures from the results of the fluid-dynamics calculations. Because cross-section shielding is based on the assumption of a homogeneous cell composition, no information about the cell structure configuration or phase composition is transmitted.

b. Quasistatic Equations. The equations used in the SIMMER-II quasistatic algorithm are given in this section. The equations are organized into five categories: (1) Flux-Shape Equations, (2) Amplitude Equation, (3) Flux Shape-Amplitude Equation Coupling terms, (4) Fluid Dynamics-Neutronics Coupling terms, and (5) Time-Step Controls. The choice of transport or diffusion theory affects categories (1) and (3); thus, both sets of equations appear.

(1) Flux-Shape Equations. The finite-difference form of the discrete-ordinates equation is derived in Appendix L, and the corresponding diffusion equation is derived in Appendix I. These equations are repeated in this section in a modified form that accounts for the solution technique used by the quasistatic approach. In the quasistatic approach, the delayed-neutron source is

treated as an inhomogeneous term. To accelerate the iterative solution of the inhomogeneous equation, the quasistatic constraint constant, γ , is used to divide the prompt-fission source. This quantity is allowed to vary during the outer iteration procedure used to converge the flux shape. When the flux is converged, γ also is converged but not necessarily to unity as required by the quasistatic method. However, recalculation of the sequence of reactivity steps and subsequent re-solution of the flux-shape equation will converge γ to unity. The equations in the following two sections are the same as Eqs. (L-16) and (I-14), respectively, except for the addition of the γ term and the dropping of the terms representing the flux-shape time dependence in the discrete-ordinates equation as described in Appendix L. However, note that the discrete-ordinates equation must be supplemented by equations relating cell-edge and cell-average fluxes before a solution can be obtained. Two alternative sets of relations (diamond-difference and weighted diamond difference) are provided in Sec. V.B.3.

Discrete-Ordinates Flux-Shape Equation. The transport equation for the flux shape is the same as that for the flux except for some additional terms. This factorization introduces one term that is added to $\Sigma_{t,ijg}$ on the loss side of the equation and a second term that, in the finite-difference form, contributes to both the source and $\Sigma_{t,ijg}$. As described in Appendix L, the presence of the second term requires the storage of the angular flux during the shape calculation. Because this significantly increases the data storage requirements and the contribution is considered to be negligible, the second term is omitted in the difference equation but treated approximately in the rebalance equation, as described in Appendix L. The resulting time-dependent, discrete-ordinate, finite-difference equation for the flux shape is given by Eq. (L-16) except that the terms corresponding to the temporal derivative of the flux shape are omitted, and γ divides the prompt fission source.

$$\begin{aligned} & \mu_m (A_{i+1/2j} \psi_{i+1/2jmg} - A_{i-1/2j} \psi_{i-1/2jmg}) \\ & + \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j}) (\alpha_{m+1/2} \psi_{ijm+1/2g} - \alpha_{m-1/2} \psi_{ijm-1/2g}) \\ & + \eta_m B_i (\psi_{ij+1/2mg} - \psi_{ij-1/2mg}) + v_{ij} (\Sigma_{t,ijg} + \frac{1}{v_g} \frac{1}{N} \frac{dN}{dt}) \psi_{ijmg} \end{aligned}$$

$$\begin{aligned}
&= V_{ij} \left(\Sigma_{s,ijg \rightarrow g} \phi_{ijg} + \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} \right. \\
&\quad \left. + \frac{1}{\gamma} \frac{x_{pg}}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} + \frac{1}{N} \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd} \right), \quad (V-29)
\end{aligned}$$

where

- ψ_{ijmg} is the integrated neutron flux-shape function for cell (i,j), angular interval m, and energy group g;
- $N(t)$ denotes the flux amplitude function;
- V_{ij} is the volume of the mesh cell;
- V_g is the average neutron velocity in neutron energy group g;
- w_m denotes the angular quadrature weight;
- μ_m is the radial direction cosine for the angular interval m;
- η_m is the axial direction cosine for the angular interval m;
- $A_{i+1/2j}$ is the radial direction surface area at the outer radius of the spatial mesh cell (defined in Appendix L);
- $\alpha_{m+1/2}$ denotes an angular redistribution coefficient to account for streaming in a curved geometry (defined in Appendix L);
- B_i is the axial direction surface area for the spatial mesh cell (defined in Appendix L);
- $\Sigma_{t,ijg}$ denotes the total collision cross section for neutron energy group g in mesh cell (i,j);
- $\Sigma_{s,ijg' \rightarrow g}$ denotes the cross section for neutron scattering from energy group g' to energy group g in mesh cell (i,j);
- x_{pg} denotes the fraction of the prompt neutrons that are emitted in neutron energy group g;
- k_0 is the initial stationary state value of k_{eff} for the system;
- C_{ijd} is the delayed-neutron precursor concentration for precursor group d in mesh cell (i,j);
- $(\nu_p \Sigma_f)_{ijg'}$ is the product of the prompt neutron yield and the macroscopic fission cross section, Σ_f , for neutron energy group g' in mesh cell (i,j);

IGD is the number of delayed-neutron precursor groups;

χ_{dg} denotes the fraction of delayed neutrons emitted in neutron energy group g from delayed neutron precursor group d ; and

λ_d is the decay constant for delayed-neutron precursor group d .

Diffusion Flux-Shape Equation. The diffusion equation for the flux shape is the same as that for the flux except for some additional terms. This factorization introduces one term that is added to $\Sigma_{r,ijg}$ on the loss side of the equation and a second term that, in finite-difference form, contributes to the source and $\Sigma_{r,ijg}$. As derived in Appendix I, the resulting time-dependent, diffusion, finite-difference equation for the flux shape is given by Eq. (I-14) except that the prompt fission source is divided by γ , as explained in the beginning of this section.

$$\begin{aligned}
 & - 4\pi\Delta z_j \left[r_{i+1/2} \frac{D_{i+1jg} D_{ijg}}{\Delta r_i D_{i+1jg} + \Delta r_{i+1} D_{ijg}} (\phi_{i+1jg} - \phi_{ijg}) \right. \\
 & \quad \left. - r_{i-1/2} \frac{D_{i-1jg} D_{ijg}}{\Delta r_{i-1} D_{ijg} + \Delta r_i D_{i-1jg}} (\phi_{ijg} - \phi_{i-1jg}) \right] \quad (V-30) \\
 & - 2\pi(r_{i+1/2}^2 - r_{i-1/2}^2) \left[\frac{D_{ijg} D_{ij+1g}}{\Delta z_j D_{ij+1g} + \Delta z_{j+1} D_{ijg}} (\phi_{ij+1g} - \phi_{ijg}) \right. \\
 & \quad \left. - \frac{D_{ij-1g} D_{ijg}}{\Delta z_{j-1} D_{ijg} + \Delta z_j D_{ij-1g}} (\phi_{ijg} - \phi_{ij-1g}) \right] \\
 & + v_{ij} \left(\Sigma_{r,ijg} + \frac{1}{v_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{\Delta t} \frac{1}{v_g} \right) \phi_{ijg} \\
 & = v_{ij} \left(\sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} + \frac{1}{\gamma} \frac{\chi_{pg}}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} \right)
 \end{aligned}$$

$$+ \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd} + \frac{1}{V_g} \frac{1}{\Delta t} \phi_{ijg}^{k-1} \quad .$$

where

- ϕ_{ijg} is the integrated neutron flux-shape function for cell (i,j) and energy group g,
- $N(t)$ denotes the flux-amplitude function,
- C_{ijd} is the delayed-neutron precursor concentration for precursor group d in mesh cell (i,j),
- V_{ij} is the volume of mesh cell (i,j),
- V_g is the average neutron velocity in neutron group g,
- $\Sigma_{r,ijg}$ denotes the removal collision cross section for neutron energy group g in mesh cell (i,j),
- $\Sigma_{s,ijg' \rightarrow g}$ denotes the cross section for neutron scattering from energy group g' to energy group g in mesh cell (i,j),
- $(\nu_p \Sigma_f)_{ijg'}$ is the product of the prompt neutron yield and the macroscopic fission cross section for neutron energy group g' in mesh cell (i,j),
- D_{ijg} denotes the diffusion coefficient for energy group g in mesh cell (i,j),
- x_{pg} denotes the fraction of the prompt neutrons that are emitted in neutron energy group g,
- k_0 is the initial stationary-state value of k_{eff} for the system,
- IGD is the number of delayed neutron precursor groups,
- x_{dg} denotes the fraction of delayed neutrons emitted in neutron energy group g from delayed neutron precursor group d, and
- λ_d is the decay constant for delayed neutron precursor group d.

Delayed-Neutron Precursor Equation. In SIMMER-II, the delayed-neutron precursor equation, Eq. (III-257), is rewritten in multigroup form as

$$\frac{\partial C_{ijd}}{\partial t} = -\lambda_d C_{ijd} + N(t) F_{ijd}^d \quad , \quad d=1, \dots, IGD \quad . \quad (V-31)$$

F_{ijd}^d is the delayed-neutron precursor shape term:

$$F_{ijd}^d = \frac{1}{k_0} \sum_{g=1}^{IGM} (\nu_d \Sigma_f)_{ijg} \phi_{ijg} \quad ,$$

where $(\nu_d \Sigma_f)_{ijg}$ denotes the product of the delayed-neutron yield ν_d and the fission cross section, Σ_f , for neutron energy group g .

(2) Amplitude Equation. The equation for the amplitude function, N , is obtained by summing Eq. (V-29) over m ; multiplying by the initial stationary adjoint flux, ϕ_{ijg}^* ; and summing over all energy groups and all mesh cells (i,j) . A similar procedure is followed for a derivation based on the diffusion equation, except that there is no initial angular quadrature required. The resulting equation is

$$\frac{dN}{dt} = \frac{\rho - \beta}{\Lambda} N + \sum_{d=1}^{IGD} \lambda_d c_d \quad . \quad (V-32)$$

In this derivation, the reactivity, ρ ; the effective delayed-neutron fraction, β ; the generation time, Λ ; and the effective delayed-neutron precursor concentrations, c_d , are defined. These terms couple the amplitude equation to the flux-shape equation. The amplitude equation has the same form for both the transport theory option and the diffusion theory option; only the definition of the coupling terms changes. Solution of the amplitude equation also involves the simultaneous solution of the delayed-neutron source equations. This set of equations is obtained by multiplying Eq. (V-31) by $\phi_{ijg}^* \chi_{dg} V_{ij}$ (where ϕ_{ijg}^* is the initial stationary adjoint flux) and summing over all energy groups and mesh cells:

$$\frac{dc_d}{dt} = -\lambda_d c_d + \beta_d \frac{N}{\Lambda} \quad , \quad d = 1, 2, \dots, IGD \quad . \quad (V-33)$$

(3) Flux-Shape/Amplitude Coupling Terms. In SIMMER II, the discrete-ordinates equations assume isotopic scattering sources; as a result, the angular flux is used to determine only the scalar flux and the cell neutron currents. A further implication of this assumption is that the calculation of some of the

flux-shape amplitude equation coupling terms is identical for both transport and diffusion theory. These terms are as follows.

Effective Delayed-Neutron Fraction.

$$\beta_d = \frac{1}{F} \sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \phi_{ijg}^* \frac{\lambda_{dg}}{k_0} \sum_{g'=1}^{IGM} (\nu_d \Sigma_f)_{ijg'} \phi_{ijg'} ,$$

where

$$\beta = \sum_{d=1}^{IGD} \beta_d . \quad (V-34)$$

Effective Neutron Generation Time.

$$\Lambda = \frac{1}{F} \sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \frac{\phi_{ijg}^* \phi_{ijg}}{v_g} = \frac{1}{F} . \quad (V-35)$$

Effective Delayed-Neutron Precursor Concentrations.

$$c_d = \sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \phi_{ijg}^* \lambda_{dg} C_{ijd} . \quad (V-36)$$

The effective neutron source is given by,

$$F = \frac{1}{k_0} \sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \phi_g^* \left[\lambda_{pg} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} + \sum_{d=1}^{IGD} \lambda_{dg} \sum_{g'=1}^{IGM} (\nu_d \Sigma_f)_{ijg'} \phi_{ijg'} \right] . \quad (V-37)$$

Equation (V-36) is not needed formally because both the effective delayed-neutron precursor concentration, c_d , and the spatial delayed-neutron

precursor concentration, C_{ijd} , are defined by initial conditions. However, in SIMMER-II, the effective concentration and the spatial concentration are calculated by numerical methods with quite different error characteristics. Thus, they may become out of balance during a long calculation, and as a result, the quasistatic constraint integral will not converge to unity during the 7-iteration at shape recalculation steps. Equation (V-36) is used in SIMMER-II to rebalance the delayed neutron sources in the amplitude and shape equations periodically.

The reactivity is the only coupling term that depends on the choice of transport or diffusion theory. The reactivity is defined by identifying certain coupling terms from the amplitude equation. Because the amplitude equation was derived by integrating the time-dependent transport or diffusion equation, terms in the reactivity equation can be associated with terms in the transport or diffusion equation. Thus,

$$\rho = \rho_{FP} + \rho_{FD} + \rho_S + \rho_L + \rho_T \quad , \quad (V-38)$$

where ρ_{FP} is the prompt fission, ρ_{FD} is the delayed fission, ρ_S is the scatter, ρ_L is the leakage, and ρ_T is the total (associated with total cross section) contribution to the reactivity. The two fission contributions are the same in both the transport and diffusion options:

$$\rho_{FP} = \frac{1}{F} \left(\sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \phi_{ijg}^* \frac{\chi_{pg}}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} \right) \quad , \quad \text{and} \quad (V-39)$$

$$\rho_{FD} = \frac{1}{F} \left(\sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \phi_{ijg}^* \sum_{d=1}^{IGD} \frac{\chi_{dg}}{k_0} \sum_{g'=1}^{IGM} (\nu_d \Sigma_f)_{ijg'} \phi_{ijg'} \right) \quad . \quad (V-40)$$

The other contributions depend on the choice of the transport or diffusion option.

Discrete-Ordinates Reactivity Equation.

$$\rho_S = \frac{1}{F} \sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \phi_{ijg}^* \sum_{g'=1}^g \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} \quad (V-41)$$

$$\rho_T = \frac{1}{F} \sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \phi_{ijg}^* \Sigma_{t,ijg} \phi_{ijg} \quad (V-42)$$

$$\begin{aligned} \rho_L = & \frac{1}{F} \sum_{i,j} \sum_{g=1}^{IGM} \phi_{ijg}^* (A_{i-1/2j} \sum_{\mu_m > 0} w_m \mu_m \psi_{i-1/2jmg} \\ & - A_{i+1/2j} \sum_{\mu_m > 0} w_m \mu_m \psi_{i+1/2jmg} + A_{i+1/2j} \sum_{\mu_m < 0} w_m \mu_m \psi_{i+1/2jmg} \\ & - A_{i-1/2j} \sum_{\mu_m < 0} w_m \mu_m \psi_{i-1/2jmg} - B_i \sum_{\eta_m > 0} w_m \eta_m \psi_{i+1/2jmg} \\ & + B_i \sum_{\eta_m > 0} w_m \eta_m \psi_{ij-1/2mg} + B_i \sum_{\eta_m < 0} w_m \eta_m \psi_{ij+1/2mg} \\ & - B_i \sum_{\eta_m < 0} w_m \eta_m \psi_{ij-1/2mg}) \\ = & \frac{1}{F} \sum_{i,j} \sum_{g=1}^{IGM} \phi_{ijg}^* (FR_{i-1/2j} - FR_{i+1/2j} + FL_{i+1/2j} - FL_{i-1/2j} \\ & - FU_{ij+1/2} + FU_{ij-1/2} + FD_{ij+1/2} - FD_{ij-1/2}) \quad , \end{aligned} \quad (V-43)$$

\

where the cell flows FR, FL, FD, and FU are defined in Appendix L.

Diffusion Reactivity Equation.

$$\rho_S = \frac{1}{F} \sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \phi_{ijg}^* \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} \quad (V-44)$$

$$\rho_T = \frac{1}{F} \sum_{i,j} v_{ij} \sum_{g=1}^{IGM} \Phi_{ijg}^* \Sigma_{r,ijg} \phi_{ijg} \quad (V-45)$$

$$\rho_L = \frac{1}{F} \sum_{i,j} \sum_{g=1}^{IGM} \Phi_{ijg}^*$$

$$[-2\pi r_{i+1/2} \Delta z_j D_{ijg} \left(\frac{\partial \phi_g}{\partial r}\right)_{r_{i+1/2}} + 2\pi r_{i-1/2} \Delta z_j D_{ijg} \left(\frac{\partial \phi_g}{\partial r}\right)_{r_{i-1/2}}$$

$$- 2\pi r_i \Delta r_i D_{ijg} \left(\frac{\partial \phi_g}{\partial z}\right)_{z_{j+1/2}} + 2\pi r_i \Delta r_i D_{ijg} \left(\frac{\partial \phi_g}{\partial z}\right)_{z_{j-1/2}}]$$

$$= \frac{1}{F} \sum_{i,j} \sum_{g=1}^{IGM} \Phi_g^* [R_{i+1j}(\phi_{i+1} - \phi_i) + L_{i-1j}(\phi_{i-1} - \phi_i) + U_{j+1}(\phi_{j+1} - \phi_j)$$

$$+ D_{ij-1g}(\phi_j - \phi_{j-1})] \quad , \quad (V-46)$$

where the finite-difference coefficients R, L, U, and D are defined in Appendix I. Each reactivity component (that is, leakage, removal, downscatter, prompt fission, and delayed fission) is evaluated separately to aid in the analysis of neutronics effects.

(4) Fluid-Dynamics/Neutronics Coupling Terms. After the fluid-dynamics calculation is interrupted for a neutronics reactivity step, the total macroscopic densities and mass-averaged temperatures needed for the neutronics calculation are determined from the appropriate fluid-dynamics quantities for each of the five components (fertile fuel, fissile fuel, steel, sodium, and control). The following equations are used for the density calculations.

Fertile fuel.

$$\bar{\rho}_{\text{Fert}} = \bar{\rho}_{s1} + \bar{\rho}_{s3} + \bar{\rho}_{\ell 1} + \bar{\rho}_{\ell 6} + \bar{\rho}_{g1} \quad (V-47)$$

Fissile fuel.

$$\bar{p}_{\text{Fiss}} = \bar{p}_{s2} + \bar{p}_{s4} + \bar{p}_{l2} + \bar{p}_{l7} + \bar{p}_{g2} \quad (\text{V-48})$$

Steel.

$$\bar{p}_{\text{SS}} = \bar{p}_{s5} + \bar{p}_{s6} + \bar{p}_{l8} + \bar{p}_{g3} \quad (\text{V-49})$$

Sodium.

$$\bar{p}_{\text{Na}} = \bar{p}_{l4} + \bar{p}_{g4} \quad (\text{V-50})$$

Control.

$$\bar{p}_{\text{Cont}} = \bar{p}_{s7} + \bar{p}_{l5} + \bar{p}_{g5} \quad (\text{V-51})$$

The fission gas component is not considered in the neutronics calculations because the effect of fission gas is negligible. The following set of equations is used for the component temperature calculations.

Fertile fuel.

$$T_{\text{Fert}} = \frac{\bar{p}_{s1} T_{S1} + \bar{p}_{s3} T_{S2} + \bar{p}_{l1} T_{L1} + \bar{p}_{l6} T_{L5} + \bar{p}_{g1} T_G}{\bar{p}_{\text{Fert}}} \quad (\text{V-52})$$

Fissile fuel.

$$T_{\text{Fiss}} = \frac{\bar{p}_{s2} T_{S1} + \bar{p}_{s4} T_{S2} + \bar{p}_{l2} T_{L1} + \bar{p}_{l7} T_{L5} + \bar{p}_{g2} T_G}{\bar{p}_{\text{Fiss}}} \quad (\text{V-53})$$

Steel.

$$T_{SS} = \frac{\bar{p}_{s5} T_{S3} + \bar{p}_{s6} T_{S4} + \bar{p}_{L3} T_{L2} + \bar{p}_{L8} T_{L6} + \bar{p}_{g3} T_G}{\bar{p}_{SS}} \quad (V-54)$$

Sodium.

$$T_{Na} = \frac{\bar{p}_{L4} T_{L3} + \bar{p}_{g4} T_G}{\bar{p}_{Na}} \quad (V-55)$$

Control.

$$T_{Cont} = \frac{\bar{p}_{s7} T_{SS} + \bar{p}_{L5} T_{L4} + \bar{p}_{g5} T_G}{\bar{p}_{Cont}} \quad (V-56)$$

3. Discrete-Ordinates Method. The neutron-transport difference equations and the methods used to solve them are similar to those used in TWOTRAN-II; likewise, the diffusion difference equations and solution methods are similar to those of DIF3D. However, time-dependent equations are being solved rather than stationary equations. As a result, eigenvalue calculations for the initial state and inhomogeneous calculations for the transient states are performed. In both cases, the same calculative modules are used with relatively minor changes made to add terms representing the time dependence.

a. Time-Dependent Discrete-Ordinates Equations. Solving Eq. (V-29) requires additional equations relating the cell boundary angular fluxes and the cell-averaged flux. Two alternative sets are provided in SIMMER-II. The "Diamond" difference equations [Eq. (L-17)] provide second-order spatial truncation error²¹ but may result in negative fluxes. If they occur, negative fluxes are treated by a "fixup" algorithm. The "Weighted-Diamond" difference equations [(Eqs. L-17c) and (L-23)] guarantee a nonnegative solution but require a somewhat greater computational effort and may result in a greater truncation

error. Use of the diamond or weighted-diamond equations to solve Eq. (V-29) results in the following equations for the angular fluxes.

"Diamond"-Difference.

$$\psi_{ijmg} = \frac{\beta_{ijmg}}{\gamma_{ijmg}} \quad , \quad (V-57)$$

where

$$\begin{aligned} \beta_{ijmg} = & (AA)_{ijm} \psi_{i\mp 1/2jmg} + (BB)_{im} \psi_{ij\mp 1/2mg} \\ & + (CC)_{ijm} \psi_{ijm-1/2g} + v_{ij} (\Sigma_{s,ijg \rightarrow g} \phi_{ijg} \\ & + \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} + \frac{1}{\gamma} \frac{x_{pg}}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} \\ & + \frac{1}{N} \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd}) \end{aligned}$$

$$\text{and } \gamma_{ijmg} = (AA)_{ijm} + (BB)_{im} + (CC)_{ijm} + \left(\Sigma_{t,ijg} + \frac{1}{v_g} \frac{1}{N} \frac{dN}{dt} \right) v_{ij} \quad .$$

The following definitions have been used in Eq. (V-57).

$$(AA)_{ijm} = |\pm \mu_m| (A_{i+1/2j} + A_{i-1/2j}) \quad ,$$

$$(BB)_{im} = 2|\pm \eta_m| B_i \quad , \text{ and}$$

$$(CC)_{ijm} = \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j})(\alpha_{m+1/2} + \alpha_{m-1/2}) \quad .$$

In the above, the purpose of including the sign of μ_m and η_m within absolute value operators is to clarify which values of μ_m and η_m (positive or negative) are associated with which ψ 's and coefficient terms. Equation (V-57) is similar to Eq. (L-21) except that the terms involving the angular flux shape from the previous time step have been omitted. See Eq. (L-38) and the accompanying explanation for information about the actual calculative expressions used in the SIMMER-II solution.

"Weighted-Diamond"-Difference.

$$\psi_{ijmg} = \frac{\beta_{ijmg} + a \alpha_{ijmg}}{\gamma_{ijmg} + a \delta_{ijmg}} \quad , \quad (V-58)$$

where β_{ijmg} and γ_{ijmg} are defined as in Eq. (V-57) above, and

$$\begin{aligned} \alpha_{ijmg} = & |\pm\mu_m| A_{i\mp 1/2j} \psi_{i\mp 1/2jmg} + \frac{(BB)_{im}}{2} \psi_{ij\mp 1/2mg} + (CC)_{ijm} \psi_{ijm-1/2g} \\ & + V_{ij} (\Sigma_{s,ijg \rightarrow g} \phi_{ijg} \\ & + \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} + \frac{1}{\gamma} \frac{x_{pg}}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} \\ & + \frac{1}{N} \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd}) \quad , \end{aligned}$$

and

$$\delta_{ijmg} = |\pm\mu_m| A_{i\mp 1/2j} + \frac{(BB)_{im}}{2} + (CC)_{ijm} + v_{ij} \left(\sum_t, ijg + \frac{1}{v_g} \frac{1}{N} \frac{dN}{dt} \right) .$$

In Eq. (V-58), if $a = 0$, the weighted-diamond solution equals the diamond solution. If using the diamond formula would produce a negative solution a nonzero "a" must be used. In Appendix L, a scheme for determining an appropriate value for a is described.

Selection of the "Diamond" or "Weighted-Diamond" options depends on the following considerations. In the "Diamond" option, when sources are positive, negative fluxes are prevented by using a "set-flux-to-zero-and-correct" approach. This scheme is most accurate because it is the minimum deviation from the second-order accuracy of the "Diamond" difference equations. All linear difference schemes that lead to positive coefficients in extrapolations are first-order accurate,²¹ and all corrector schemes other than the set-to-zero scheme lead to positive extrapolated fluxes, that is, to a greater deviation from the negative diamond-difference result. In addition, the set-to-zero scheme is more selective, not only because it is used only when a flux is negative, unlike a weighted diamond scheme but also because it is applied to the extrapolations individually, unlike a scheme in which all fluxes are recomputed with a step function approximation.

b. Fine-Mesh Rebalance Equations. Convergence of the iterative solution of the angular flux can be accelerated by enforcing cell-wise neutron conservation for each inner iteration. The deviation from neutron conservation is determined by calculating the neutron flows into and out of each cell. The scalar-flux shape for each cell then is multiplied by a factor that results in neutron conservation, that is,

$$\phi_{ijg}^c = f_{ij} \phi_{ijg}^u \quad , \quad (V-59)$$

where

ϕ_{ijg}^c is the corrected flux shape,

ϕ_{ijg}^u is the uncorrected flux shape, and

f_{ij} is the correction factor for the flux in cell (ij) in group g.

A rebalance equation is derived by equating losses in the cell (outflow plus absorption) to the sources (scattering sources plus inflow from adjoining cells), as described in Appendix L [see Eq. (L-37)]:

$$\begin{aligned}
 & f_{ij} [FR_{i+1/2j}^u + FL_{i-1/2j}^u + FU_{ij+1/2}^u + FD_{ij-1/2}^u \\
 & + (\Sigma_{t,ijg} + \frac{1}{V_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{\Delta t} \frac{1}{V_g}) V_{ij} \phi_{ijg}] \\
 & - f_{i+1j} FL_{i+1/2j}^u - f_{i-1j} FR_{i-1/2j}^u - f_{ij+1} FD_{ij+1/2}^u - f_{ij-1} FU_{ij-1/2}^u \\
 & = (\Sigma_{s,ijg} \phi_{ijg} + \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'}) \\
 & + \frac{1}{\gamma} \frac{\chi_{pg}}{k_o} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} + \frac{1}{N} \sum_{d=1}^{IGD} \chi_{dg} \lambda_d C_{ijd} + \frac{1}{\Delta t} \frac{1}{V_g} \phi_{ijg}^{k-1} V_{ij} .
 \end{aligned} \tag{V-60}$$

In Eq. (V-60), ϕ_{ijg}^{k-1} is the scalar flux shape from the previous time step. Solution of Eq. (V-60) for the rebalance factors and modification of the scalar fluxes by use of Eq. (V-59) yields a flux shape that satisfies the neutron conservation requirement. This process is performed once each scattering source iteration until the within-group scattering source has converged.

c. Boundary Conditions. The neutronics boundary conditions in SIMMER-II are fixed, and no options are available. For the discrete-ordinates equations, a reflective boundary condition is used at the left boundary. The value of the flux on the boundary for incoming directions is set equal to the value of the outgoing flux in the direction corresponding to specular reflection. If the leftmost cell of the neutronics mesh is not bounded by the centerline, a constant flux level is modeled in the interior annulus. For the top, bottom, and right boundaries, a vacuum boundary condition is used; that is, the value of the angular flux on the boundary is set to zero for all incoming directions.

For the rebalance equations, the boundary conditions for rebalance factors relate to the scalar flux; consistency with the discrete-ordinate equation implies vacuum boundary conditions on the top, right, and bottom of the system. The corresponding inflows are zero, thus eliminating the appropriate f 's from the equation. On the left side of the system, a reflective boundary condition exists so that $f_{i-1,j} = f_{ij}$. With these boundary conditions, Eq. (V-60) can be solved for the rebalance factors, f_{ij} .

d. Adjoint Stationary Equations. The quasistatic method requires a weighting function to calculate the amplitude equation parameters. Normally, this weighting function is taken as the adjoint flux for the stationary reactor state before initiating the accident. However, for SIMMER-II calculations that begin from transient reactor states predicted by other computer programs, the stationary reactor state is not available. The stationary adjoint flux for the transient reactor state at the beginning of the SIMMER-II calculation is considered to be a reasonably good weighting function that does not differ significantly from the normal adjoint-flux weighting function. This stationary adjoint flux is the solution to the adjoint equation:

$$\begin{aligned}
 & -\mu_m (A_{i+1/2j} \psi_{i+1/2jmg}^* - A_{i-1/2j} \psi_{i-1/2jmg}^*) \\
 & - \frac{1}{w_m} (A_{i+1/2} - A_{i-1/2j}) (\alpha_{m+1/2} \psi_{ijm+1/2g}^* - \alpha_{m-1/2} \psi_{ijm-1/2g}^*) \\
 & - \eta_m B_i (\psi_{ij+1/2mg}^* - \psi_{ij-1/2mg}^*) + V_{ij} \Sigma_{t,ijg} \psi_{ijmg}^* \\
 & = V_{ij} \left(\sum_{g'=g}^{IGM} \Sigma_{s,ijg \rightarrow g'} \phi_{ijg'}^* + \frac{(\nu_p \Sigma_f)_{ijg}}{k'_0} \sum_{g'=1}^{IGM} x_{pg'} \phi_{ijg'}^* \right. \\
 & \left. + \sum_{d=1}^{IGD} \frac{(\nu_d \Sigma_f)_{ijg}}{k'_0} \sum_{g'=1}^{IGM} x_{dg'} \phi_{ijg'}^* \right), \tag{V-61}
 \end{aligned}$$

where

Ψ_{ijmg}^* denotes the adjoint flux for cell (i,j), angular interval m, and energy group g,

Φ_{ijg}^* denotes the scalar adjoint flux for cell (i,j) and energy group g, and

k'_0 is the stationary value of k_{eff} for the adjoint system.

Boundary conditions for Eq. (V-61) are as follows. A reflective boundary condition is used at the left boundary. The value of the adjoint flux on the boundary for incoming direction is set equal to the value of the outgoing flux in the direction corresponding to specular reflection. For the top, bottom, and right boundaries, the value of the angular flux on the boundary is set to zero for all outgoing directions. With these boundary conditions, the adjoint eigenvalue, k'_0 , is formally equal (there may be differences because of different numerical characteristics) to the eigenvalue solving the real stationary transport equation. Equation (V-61) differs from the real stationary equation in the following respects: (1) the sign of the divergence terms is negative, (2) the scattering source is transposed with respect to energy, and (3) neutronics parameters of the fission source have been interchanged.

The divergence terms in Eq. (V-61) are multiplied by -1 and thus will correspond to the divergence terms in the real stationary equation if the solution Ψ_{ijmg}^* is associated with the direction $-\vec{\Omega}_m$. This approach has been followed with SIMMER-II and has no effect because only the scalar adjoint flux, which is not affected, is used. Note that the adjoint flux is evaluated only once (at the beginning of the SIMMER-II calculation) and not at every neutronics time step. A time-dependent weighting function would change the quasistatic-method equations.

As with the discrete-ordinates equations for the real flux, either the "Diamond" difference or the "Weighted-Diamond" difference relations supplement Eq. (V-61) to define the adjoint flux. Those equations are unchanged from the representation appearing in Appendix L, except that the adjoint flux is substituted for the real flux.

4. Diffusion Theory Method.

a. Time-Dependent Diffusion Equation. The diffusion equation used in SIMMER-II is derived in Appendix I and is restated as Eq. (V-30). Expressing Eq. (V-30) as a matrix equation yields the familiar five-diagonal form. The main diagonal contains terms describing the rate of change, group removal, and cell leakages. The super and outer diagonals contain radial and axial leakage terms, respectively. Because the leakage across common cell boundaries is iden-

tical for adjacent cells, the matrix is symmetric. The detailed matrix structure is given in Appendix I.

Solution of the matrix equation proceeds as described in Sec. V.C and is similar to the solution of rebalance factors used for accelerating the transport solution. Because the matrix coefficients do not change during the outer iteration, they are calculated simultaneously with the macroscopic cross sections upon which they depend. The coefficients then are read into SCM (from LCM or disk) as required for each inner iteration.

b. Boundary Conditions. The left boundary is treated as a reflective boundary,

$$\left. \frac{\partial \phi}{\partial r} \right|_{r=r_i} = 0 \quad .$$

The user can place the left boundary to the right of the axis of symmetry. In this case, an annulus with a constant central region flux level is modeled. The right, bottom, and top boundaries are treated as vacuum boundaries,

$$\phi(r = r_{\max}) = \phi(z = z_{\min}) = \phi(z = z_{\max}) = 0 \quad .$$

c. Adjoint Stationary Equation. The quasistatic method requires a weighting function to calculate the amplitude-equation parameters. For both the diffusion and transport options, a stationary solution of the adjoint equation is used for a weighting function. The adjoint diffusion equation solved by SIMMER-11 is given by,

$$\begin{aligned} & - 4\pi \Delta z_j \left[r_{i+1/2} \frac{D_{i+1jg} D_{ijg}}{\Delta r_i D_{i+1jg} + \Delta r_{i+1} D_{ijg}} (\phi_{i+1jg}^* - \phi_{ijg}^*) \right. \\ & \left. - r_{i-1/2} \frac{D_{i-1jg} D_{ijg}}{\Delta r_{i-1} D_{ijg} + \Delta r_i D_{i-1jg}} (\phi_{i-1jg}^* - \phi_{ijg}^*) \right] \end{aligned}$$

$$\begin{aligned}
& - 2\pi (r_{i+1/2}^2 - r_{i-1/2}^2) \left[\frac{D_{ijg} D_{ij+1g}}{\Delta z_j D_{ij+1g} + \Delta z_{j+1} D_{ijg}} (\Phi_{ij+1g}^* - \Phi_{ijg}^*) \right. \\
& \left. - \frac{D_{ij-1g} D_{ijg}}{\Delta z_{j-1} D_{ijg} + \Delta z_j D_{ij-1g}} (\Phi_{ijg}^* - \Phi_{ij-1g}^*) \right] + \Sigma_{r,ijg} \Phi_{ijg}^* V_{ij} \\
& = V_{ij} \left(\frac{(\nu_p \Sigma_f)_{ijg}}{k'_0} \sum_{g'=1}^{IGM} x_{pg'} \Phi_{ijg'}^* + \sum_{d=1}^{IGD} \frac{(\nu_d \Sigma_f)_{ijg}}{k'_0} \sum_{g'=1}^{IGM} x_{dg'} \Phi_{ijg'}^* \right) . \quad (V-62)
\end{aligned}$$

Boundary conditions are the same as for the real diffusion equation; that is, the left boundary is reflective and the right, bottom, and top boundaries are vacuum. As in the transport option, the adjoint flux is evaluated only once at the beginning of the transient.

5. Delayed-Neutron Precursors. In SIMMER-II, the delayed-neutron yield for each delayed-neutron group is assumed to be a constant, β_d (supplied by the user), times the total neutron yield; that is, $\nu_{dg} = \beta_d \nu_g$. Integration of Eq. (V-31) leads to

$$C_{ijd} = \int_0^t N(\tau) F_{ijd}^d e^{-\lambda_d(t-\tau)} d\tau + C_{0,ijd} e^{-\lambda_d t} , \quad (V-63)$$

where $C_{0,ijd}$ is the initial precursor concentration. In SIMMER-II it is assumed that the time dependence of the delayed-neutron source is given by

$$F_{ijd}^d(t) = F_{ijd}^d(t_1) \left(1 - \frac{t}{t_2 - t_1}\right) + F_{ijd}^d(t_2) \left(\frac{t}{t_2 - t_1}\right) , \quad (V-64)$$

where t_1 and t_2 denote two subsequent times at which the flux shape is recalculated and t is relative to t_1 . For simplicity, the spatial dependence of the delayed neutron source is not shown. Substitution of Eq. (V-64) into (V-63) gives

$$C_{ijd}(t) = A_{h,d} F_{ijd}^d(t_1) + B_{h,d} F_{ijd}^d(t_2) + C_{o,ijd} e^{-\lambda_d t} \quad , \quad (V-65)$$

where

$$A_{h,d} = \int_0^t \left(1 - \frac{\tau}{t_2 - t_1}\right) N(\tau) e^{-\lambda_d(t-\tau)} d\tau \quad , \text{ and}$$

$$B_{h,d} = \int_0^t \frac{\tau}{t_2 - t_1} N(\tau) e^{-\lambda_d(t-\tau)} d\tau \quad .$$

Evaluation of the integrals determining $A_{h,d}$ and $B_{h,d}$ is described in Appendix M. Calculation of C_{ijd} at the different levels of time step required by the quasistatic method is described in Sec. V.C.2.

C. Solution Algorithms

The methods used to solve the time-dependent transport or diffusion shape equations are essentially the same whether the solution is a homogeneous (eigenvalue) calculation for the initial state or an inhomogeneous calculation for a transient state. The only change is to include terms that represent the time-dependence in the quasistatic treatment. Likewise, there are similarities between the transport and diffusion equation solution approaches as described below.

In some cases, the desired initial neutronics state may not be the pretransient stationary state. For example, if SIMMER-II is used to continue calculations begun by an initiating-phase accident code such as SAS3D, the neutronics calculations must be initialized for a nonstationary reactor state. The method developed for the transient initialization uses quasistatic methodology to give consistency between the (constant) point-kinetics parameters used for the initiating-phase calculation and the (time-dependent) amplitude-equation parameters of SIMMER-II. As a result, reactivity, power, and other integral kinetics parameters remain continuous at the transition between the two codes. Details of the transient initialization are provided in Appendix H.

1. Flux-Shape Solution. The solution method for the flux equations has two levels of iteration. The first level, the outer iteration, consists of computing the downscatter, fission, and delayed-neutron sources given by the last three terms, respectively, on the right side of Eq. (V-29) in the case of the

transport option or Eq. (V-30) in the case of the diffusion option, based on the fluxes of the previous iteration. The fission source eigenvalue, k_0 , is adjusted after each iteration until a stationary solution is obtained. The outer iteration procedure is the same for both the transport and diffusion equations. The second level, the inner iteration, is different in transport and diffusion theory. In transport theory, the inner iteration involves computing the within-group scattering source and then solving for the flux. The within-group source is adjusted after each inner iteration until a stationary solution is obtained. In diffusion theory, only the flux solution is involved in the inner iteration algorithm because the source is fixed.

a. Inner Iterations for the Discrete-Ordinates Method. For the discrete-ordinates method, the group-flux calculation is performed after the total group source is determined. The fission, downscatter, and delayed-neutron sources are determined in the outer iteration. Computation of the in-group scattering source is part of the inner iteration because it depends on the unknown group flux. Therefore, the inner iteration consists of a cyclic determination of the within-group scattering source based on the group flux from the previous inner iteration, followed by a computation of the group flux for the current iteration. The iteration is terminated when a stationary solution is achieved. The following calculative sequence is used.

1. Calculate the within-group sources for both the discrete ordinates and rebalance equations.
2. Perform two downward space-angle traverses.
 - A. Perform sweep 1 (inward space-angle traverse)
 - B. Calculate the left horizontal flows
 - C. Perform sweep 2 (outward space-angle traverse)
 - D. Calculate the right horizontal flows
 - E. Calculate the downward vertical flows
3. Perform two upward space-angle traverses.
 - A. Perform sweep 3 (inward space-angle traverse)
 - B. Calculate the left horizontal flows
 - C. Perform sweep 4 (outward space-angle traverse)
 - D. Calculate the right horizontal flows
 - E. Calculate the upward vertical flows
4. Calculate cell loss terms for the rebalance equation.
5. Adjust the flows for rebalance stability.

6. Calculate and apply the rebalance factors.
7. Check convergence and return to step 1 if necessary.
8. Calculate the group contribution to the quasistatic constraint integral.
9. Calculate the group transport leakage contribution to the reactivity.

The unknowns, ψ , are ordered so that the difference scheme is stable and the coefficient matrix is lower triangular. Physically, this corresponds to moving in the direction of particle flow.

Computation begins for the $\mu < 0$, $\eta < 0$ quadrant of directions at the top right corner of the system. The traverse through the entire space mesh for these directions is referred to as Sweep 1. The sweep begins with storing the right boundary condition and computing the flow through the boundary. Sweep 1 begins first for $J = JT$, that is, along the top boundary of the system. For this value of J , I runs from the right: $IT, IT-1, \dots, 1$. For each I , M runs in the sequence shown, for example, in Figs. (L-2) and (L-3) before the next value of I is taken. The angular flux shape ψ_{ijmg} is computed in the following calculative expressions, which are repeated (except for adding the quasistatic constraint constant to the source term) from Appendix L [Eqs. (L-38) and (L-39)].

"Diamond"-Difference.

$$\psi_{ijmg} =$$

$$\frac{(AA)'_{ijm} \psi_{i-1/2jmg} + (BB)'_{im} \psi_{ij-1/2mg} + (CC)'_{ijm} \psi_{ijm-1/2g} + \frac{1}{\gamma} S_{c,ijg}}{(AA)'_{ijm} + (BB)'_{im} + (CC)'_{ijm} + (CT)_{ijg}} \quad (V-66)$$

"Weighted-Diamond" Difference.

$$\psi_{ijmg} = \frac{N_W}{D_W} \quad (V-67)$$

where

$$N_W = (AW*ALPH + BETAP) \quad \text{and}$$

$$D_W = (AA)'_{ijm} + (BB)'_{ijm} + (CC)'_{ijm} + (CT)_{ijg} \\ + AW* ((CT)_{ijg} + (CC)'_{ijm} + A(LA4)*A(LCM) + 1/2(BB)'_{ijm}) \quad ,$$

where the terms used are defined in Appendix L. After Eq. (V-66) or (V-67) is evaluated, the angular fluxes are calculated from the "Diamond" difference relations [Eq. (L-17)] or the "Weighted-Diamond" difference relations [Eqs. (L-17c) and (L-23)], respectively. For a direction for which there is no value for $\psi_{ijm-1/2g}$, Eqs. (V-66) or (V-67) are used with $(CC)'_{ijm} = 0$. If the "Diamond"-Difference relations are used and if the computed angular fluxes are negative, a "set-flux-to-zero-and-correct" approach is implemented (discussed below), and new angular flux values are computed. The angular fluxes then are used to accumulate the scalar flux sum using

$$\phi_{ijg} = \sum_{\substack{\mu_m < 0 \\ \eta_m < 0}} w_m \psi_{ijmg} \quad . \quad (V-68)$$

This same sum accumulates in all sweeps. The contribution to the partial flow also is computed.

When the flux computation has been completed for all values of "m," the next I cell to the left is computed for the current J using the same process. The flux extrapolated in the previous cell is now properly the right boundary condition. When the inside boundary is reached, the flux has been computed for all downward-pointing incoming directions along the top interval of the system. The value of the extrapolated flux $\psi_{ijm+1/2g}$ for each space cell has been saved for each of the ISN η levels corresponding to the $\mu = 0$ value. These values initialize $\psi_{ijm-1/2g}$ for Sweep 2 that now begins for the current (top) J level. The left boundary condition is set, and Sweep 2 proceeds along the top interval of the system, from left to right for all downward-outward directions. When the right boundary is reached, Sweep 1 for the next lower J interval is begun. The cell edge fluxes calculated in the cell above are now stored for both inward and outward directions, and computation proceeds just as it did for the top interval. This procedure continues first inward, then outward, for each J level for decreasing values of J until the bottom of the system is reached. The

bottom boundary condition then is set, and the calculation moves upward in J, first inward (Sweep 3) and then outward (Sweep 4) on each J level. When the top is reached, all new fluxes and partial flows have been generated. At this point, the one space-angle traverse for this group is complete.

Although all the coefficients of Eq. (V-66) are positive, extrapolations of the form $\psi_{i+1/2jmg} = 2\psi_{ijmg} - \psi_{i-1/2jmg}$ can lead to negative values for fluxes. When sources are positive, negative fluxes are prevented by using a "set-flux-to-zero-and-correct" approach.

The logic of the set-to-zero scheme is that if any one flux is negative, it is set to zero and the cell-centered flux, ψ_{ijmg} , is recomputed assuming that particular flux is zero. If this ψ is negative, the correction attempt is aborted. If not, the other two fluxes are reextrapolated. If either one of these is negative, ψ_{ijmg} is recomputed assuming two fluxes are negative. If this ψ_{ijmg} is positive, the remaining flux is extrapolated. If it is negative, then ψ_{ijmg} is recomputed assuming all three boundary fluxes are zero (provided the total cross section is not zero). If any ψ_{ijmg} is negative (implying a negative source or boundary flux) or if any correction leads to a division by zero (as in voids), the correction attempt is aborted.

Formulas for the correction are derived in the following manner. Suppose $\psi_{i+1/2jmg}$ is being calculated and is found to be negative. Equation (L-17) then is used in Eq. (V-29), where $\psi_{i+1/2jmg}$ is taken to be zero. Then,

$$\psi_{ijmg} = \frac{N_F}{N_D} \quad , \quad (V-69)$$

where

$$\begin{aligned} N_F = & \mu_m A_{i-1/2j} \psi_{i-1/2jmg} + 2 \eta_m B_i \psi_{ij-1/2mg} \\ & + \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j}) (\alpha_{m+1/2} + \alpha_{m-1/2}) \psi_{ijm-1/2g} \\ & + v_{ij} \frac{1}{\gamma} S'_{ijg} \quad \text{and} \end{aligned}$$

$$N_D = \frac{2}{w_m} (A_{i+1/2} - A_{i-1/2}) \alpha_{m+1/2} + 2 \eta_m B_i + v_{ij} (\Sigma_{t,ijg} + \frac{1}{v_g} \frac{1}{N} \frac{dN}{dt}) .$$

The source, S'_{ijg} , is defined for Eq. (L-12). Equation (V-69) is related closely to Eq. (V-57), as can be seen from Eqs. (L-18) through (L-21). Computationally, Eq. (V-69) is treated as follows.

$$\psi_{ijmg} = \frac{N_{Fc}}{N_{Dc}} , \quad (V-70)$$

where $N_{Fc} = A(LCM)*A(LA4)*\psi_{i-1/2jmg} + (BB)'_{im} * \psi_{ij-1/2mg} + (CC)'_{ijm} * \psi_{ijm-1/2g}$

$$+ \frac{1}{7} S_{c,ijg} \text{ and}$$

$$N_{FD} = 2*A(LA2)*A(LAL2) + (BB)'_{im} + (CT)_{ijg} .$$

The terms used in Eq. (V-70) are defined in Appendix L [Eq. (L-38)]. If $\psi_{i-1/2jmg}$ were being computed and found negative, then Eq. (V-70) would be altered by using $A_{i+1/2j} \psi_{i+1/2jmg}$ instead of $A_{i-1/2j} \psi_{i-1/2jmg}$. In either case, the negative flux is set to zero, and, provided ψ_{ijmg} is positive, the remaining two fluxes, $\psi_{ij-1/2mg}$ and $\psi_{ijm+1/2g}$, are computed from Eq. (L-17). All other necessary correction formulas (six more) can be generated in the same fashion. These appear explicitly in subroutine FIXUP in the program.

A stability problem can occur whenever the mesh cells are optically thick with a scattering ratio near unity. One criterion²² for rebalance stability is

$$\frac{(2c - 1)\Sigma_{t,ijg} \Delta x}{4} < \frac{D_{ijg}}{\Delta x} , \quad (V-71)$$

where

c is the scattering ratio ($\Sigma_{s,ijg}/\Sigma_{t,ijg}$),

Δx is the mesh cell thickness (Δr_i or Δz_j), and

D_{ijg} is the diffusion coefficient.

If D is small or Δx is large and the above inequality is not satisfied, the rebalance method will be unstable.

To circumvent the instability problem, a smoothing technique suggested in Ref. 22 was modified and implemented in SIMMER-II. The smoothing technique consists of modifying the partial flows from a mesh cell. For the (r,z) geometry in SIMMER-II, a factor must be calculated for the z direction as

$$\gamma_{ij+1/2} = \left[\frac{\Sigma_{s,ijg} \Delta z_j}{4} \max (\phi_{ijg}, \phi_{ij+1g}) \right] B_i \quad , \quad (V-72)$$

and for the r direction as

$$\gamma_{i+1/2j} = \left[\frac{\Sigma_{s,ij} \Delta r_i}{4} \max (\phi_{ijg}, \phi_{i+1jg}) \right] A_{i+1/2j} \quad . \quad (V-73)$$

The $\gamma_{ij+1/2}$ values calculated using Eq. (V-72) are added to the upward (FU) and downward (FD) flows, and the $\gamma_{i+1/2j}$ values from Eq. (V-73) are added to the rightward (FR) and leftward (FL) flows. The modified flows then are used in the rebalance algorithm.

In summary, the inner iteration consists of the following steps.

- (1) Form the within-group scatter source for the group.
- (2) Beginning with the top row of mesh cells, calculate the downward and inward fluxes moving from right to left.
- (3) Complete the downward fluxes by calculating the outward fluxes moving from left to right.
- (4) Repeat steps (2) and (3), moving down a row of mesh cells each time until the downward fluxes for the bottom row of mesh cells are calculated.
- (5) Beginning with the bottom row of mesh cells, calculate the upward and inward fluxes, moving from right to left.
- (6) Complete the upward fluxes by calculating the outward fluxes moving from left to right.
- (7) Repeat steps (4) and (5), moving up a row of mesh cells each time until the upward fluxes for the top row of mesh cells are calculated.

- (8) Adjust the partial flows to avoid instabilities in the rebalance algorithm.
- (9) Calculate the rebalance factors and apply them to the fluxes.
- (10) If the inner iteration has not converged, return to step 1 using the most recently calculated group flux. Otherwise, the iteration is complete.

In each of the sweeps of steps (2), (3), (5), and (6), the negative flux fixup is performed if needed for the "diamond" difference option. In addition, the partial flows necessary for the rebalance factor calculation are obtained.

The inner iteration is terminated when the relative difference in the cell-dependent group flux is less than a variable convergence precision, ϵ_{IN} . Only cell-dependent fluxes greater than half the maximum flux are compared because small fluxes may be encountered on the problem boundary. The variable convergence precision is dependent on the estimated error in the eigenvalue for the outer iteration and on the two input variables, EPSFAC and EPSMIN, such that

$$\epsilon_{IN} = \begin{cases} 10^{-3} & \text{if } \frac{\epsilon}{\text{EPSFAC}} > 10^{-3} \\ \frac{\epsilon}{\text{EPSFAC}} & \text{if } 10^{-3} \geq \frac{\epsilon}{\text{EPSFAC}} \geq \text{EPSMIN} \\ \text{EPSMIN} & \text{if } \frac{\epsilon}{\text{EPSFAC}} < \text{EPSMIN} \end{cases} \quad (\text{V-74})$$

where $\epsilon = \left| \frac{k^{L-1}}{k_0^L} - 1 \right|$ is the estimated error in the integrated fission source for the current outer iteration.

b. Inner Iterations for the Diffusion Theory Method. The method for solving the neutron diffusion equation is similar to the one used to solve the transport equation. The outer iteration remains unchanged. The inner iteration is simplified by eliminating the space-angle sweep and the scattering source iteration because these are not involved in diffusion theory. Because the diffusion equation is identical in structure to the rebalance equation, the solution algorithm for the rebalance equation is applied directly to the diffusion equation. The inner iteration for solution of the diffusion equation involves the following steps.

1. Add the group contribution of the shape function time derivative term to the group source.
2. Add the group contribution of the shape function time derivative term and the inverse period term to the group loss expression.
3. Iteratively solve for the spatial group flux shape.
4. Calculate the group contribution to the quasistatic constraint integral if needed.

c. Matrix Solution Techniques for Inner Iteration Solution Methods. In

SIMMER-II, the same matrix solution method is used for both the rebalance equation solution, which is part of the discrete-ordinates inner iteration, and the solution of the diffusion equations for a given source, which comprises the diffusion theory inner iteration.

In this method, successive line overrelaxation is used with a relaxation parameter ω , and successive values for each iteration are calculated from

$$\underline{f}^n = (1 - \omega) \underline{f}^{n-1} + \omega A^{-1} (L \underline{f}^n + U \underline{f}^{n-1} + \underline{q}) \quad , \quad (V-75)$$

where $\underline{f}$ is a vector quantity denoting the rebalance factors for each cell in transport theory or the unknown diffusion fluxes for each cell in diffusion theory. Following Hageman,²³ the spectral radius of the point Jacobi iteration matrix, $A^{-1} (L + U)$, initially is estimated to be zero; hence, if the relaxation parameter for the successive line overrelaxation iteration matrix is determined from

$$\omega = \frac{2}{1 + (1 - \mu^2)^{1/2}} \quad , \quad (V-76)$$

then $\omega = 1$. The user specifies the number of iterations of Eq. (V-75) using this parameter. Then the spectral radius of the successive line overrelaxation iteration matrix is estimated as

$$\mu_{\omega}^2 = \frac{(\underline{f}^{n+1} - \underline{f}^n, \underline{f}^{n+1} - \underline{f}^n)}{(\underline{f}^n - \underline{f}^{n-1}, \underline{f}^n - \underline{f}^{n-1})} \quad . \quad (V-77)$$

A new spectral radius for the Jacobi matrix is estimated from this value as

$$\mu = \frac{\mu_{\omega} + \omega - 1}{\omega (\mu_{\omega})^{1/2}} , \quad (V-78)$$

and a new relaxation parameter, ω , is determined from Eq. (V-76). The sequence of Eqs. (V-77), (V-78), and (V-76) is repeated until convergence of the rebalance equations is obtained.

The relaxation parameter is not recalculated if μ_{ω} is greater than unity. In that case, the iterations are continued with $\omega = 1$. When this method is applied to the rebalance equations for the transport equation, the iteration matrix changes every outer iteration. Therefore, it is necessary to recalculate ω every outer iteration.

The solution of the matrix equation term involving the tridiagonal matrix A is accomplished by factoring it into the form

$$A = MN , \quad (V-79)$$

where M is a lower triangular matrix with unit diagonal elements and N is an upper triangular matrix.

The solution of the matrix equation involving A then takes advantage of the fact that M is a lower triangular matrix with unit diagonal elements and N is an upper triangular matrix. The solution equation then takes the form

$$N\underline{f} = M^{-1}\underline{s} , \quad (V-80)$$

where $\underline{s} = L\underline{f}^n + U\underline{f}^{n-1} + \underline{q}$.

The calculation of $M^{-1}\underline{s}$ and $N^{-1} (M^{-1}\underline{s})$ involves only back substitution. Because the diagonal elements of M are unity, no division operation is necessary in the first step. The diagonal elements of N are divisors in the second operation, but their inverses are precalculated and stored. This procedure eliminates all but the initial divisions in the iterative solution of Eq. (V-80).

Equation (V-75) is solved iteratively. Convergence is satisfied when

$$\sum_{i,j} (f_{ij}^{m+1} - f_{ij}^m)^2 \leq \epsilon_x^2 \sum_{i,j} (f_{ij}^{m+1})^2 ,$$

where $\epsilon_x = \epsilon_{RBL}$ if the transport option is used and $\epsilon_x = \epsilon_R$ if the diffusion option is used.

Both ϵ_{RBL} and ϵ_R are variable convergence precisions that depend on the degree of outer iteration convergence. In the early stages of the fission-source iteration, a tight convergence for the rebalance or diffusion flux solution is unnecessary, and thus ϵ_x varies to avoid unnecessary iteration. The convergence precision for the matrix iteration is loose initially and then is tightened as the fission-source iteration converges. The value of ϵ_R and ϵ_{RBL} are determined as follows.

Transport Theory.

The value of ϵ_{RBL} is determined from the estimated eigenvalue error, ϵ , in the outer iteration and the two input variables, FACRBL and EPSMIN, such that

$$\epsilon_{RBL} = \begin{cases} 10^{-3} & \text{if } \epsilon/\text{FACRBL} > 10^{-3} \\ \epsilon/\text{FACRBL} & \text{if } 10^{-3} \geq \epsilon/\text{FACRBL} \geq \text{EPSMIN} \\ \text{EPSMIN} & \text{if } \epsilon/\text{FACRBL} < \text{EPSMIN} \end{cases} \quad (\text{V-81})$$

If the iteration fails to converge after a specified number of iterations, then all f 's are set to unity.

Diffusion Theory.

The value of ϵ_R is determined from the estimated eigenvalue error, ϵ , in the outer iteration and two input variables, EPSFAC and EPSMIN, such that

$$\epsilon_R = \begin{cases} 10^{-3} & \text{if } \epsilon/\text{EPSFAC} > 10^{-3} \\ \epsilon/\text{EPSFAC} & \text{if } 10^{-3} \geq \epsilon/\text{EPSFAC} \geq \text{EPSMIN} \\ \text{EPSMIN} & \text{if } \epsilon/\text{EPSFAC} < \text{EPSMIN} \end{cases} \quad (\text{V-82})$$

If the iteration fails to converge after a specified number of iterations, the last iterate values are used for f .

In both the transport- and diffusion-theory termination conditions, the estimated error in the eigenvalue for the initial solution is given by,

$$\epsilon = \left| \frac{k^{\ell-1}}{k_0^{\ell}} - 1 \right| , \quad (V-83)$$

where ℓ is the outer iteration number. For the transient solutions, the integrated fission source is substituted for k^{ℓ} .

d. Outer Iterations. The outer iteration consists of determining a cell-dependent fission-source estimate and, in the case of the initial homogeneous flux calculation (whether transient or stationary), an eigenvalue estimate and then computing the resulting fluxes in the inner iteration. The iteration is converged when a stationary solution is obtained.

The unnormalized fission source, $\Omega_{u,ij}$, is calculated from the most recent neutron fluxes and the macroscopic fission cross sections as

$$\Omega_{u,ij}^{\ell} = \sum_{g=1}^{IGM} (\nu_p \Sigma_f)_{ijg} \phi_{ijg}^{\ell} , \quad (V-84)$$

where ν_p denotes the prompt neutron yield per fission and ℓ denotes the outer iteration number.

An estimate of the eigenvalue, k_0^{ℓ} , for the ℓ^{th} iteration is obtained from the ratio of the integrated fission sources for two iterations.

$$k_0^{\ell} = \frac{\sum_{ij} \Omega_{u,ij}^{\ell} \Omega_{ij}^{\ell-1} v_{ij}}{\sum_{ij} \Omega_{u,ij}^{\ell} \Omega_{u,ij}^{\ell} v_{ij}} , \quad (V-85)$$

where $\Omega_{u,ij}^{\ell} v_{ij}$ is used as a weighting function.

For the inhomogeneous flux solutions obtained in the transient calculation, k_0^{ℓ} represents the fission source ratio for successive iterations. For the ℓ^{th}

iteration of the homogeneous calculation, the normalized fission source is obtained by dividing by k_0^ℓ .

$$\Omega_{ij}^\ell = \frac{1}{k_0^\ell} \Omega_{u,ij}^\ell . \quad (V-86)$$

In the transient inhomogeneous calculation, the fission source is normalized by the calculated initial eigenvalue. If the standard power-iteration method is chosen, Ω^ℓ becomes the fission-source guess, $\tilde{\Omega}_{ij}^{\ell+1}$, for the next iteration such that

$$\tilde{\Omega}_{ij}^{\ell+1} = \Omega_{ij}^\ell .$$

If the Chebyshev acceleration method²⁴ is chosen, the fission source guess is determined by a linear combination of fission sources from previous iterations,

$$\tilde{\Omega}_{ij}^{\ell+1} = \alpha_\ell \Omega_{ij}^\ell + (1 - \alpha_\ell + \beta_\ell) \Omega_{ij}^{\ell-1} - \beta_\ell \Omega_{ij}^{\ell-2} , \quad (V-87)$$

where

$$\alpha_1 = \frac{2}{2 - \sigma_0} , \quad \beta_1 = 0 \text{ for } \ell = 1 , \text{ and}$$

$$\alpha_\ell = \frac{4}{\sigma_0} \left\{ \frac{\cosh [(\ell - 1) \gamma_c]}{\cosh (\ell \gamma_c)} \right\} , \quad \text{for } \ell \geq 2$$

$$\beta_\ell = \frac{\cosh [(\ell - 2) \gamma_c]}{\cosh (\ell \gamma_c)} = \alpha_\ell \left(1 - \frac{\sigma_0}{2} \right) - 1 \quad (V-88)$$

In Eq. (V-88),

$$\gamma_c = \cosh^{-1} \left(\frac{2}{\sigma_0} - 1 \right) , \quad (V-89)$$

and the eigenvalue dominance ratio, σ_0 , is defined by

$$\sigma_0 = \max_{i \neq 1} \frac{|\lambda_i|}{\lambda_1} , \quad (V-90)$$

where λ_i denotes the eigenvalues of the iteration matrix; that is, the matrix that multiplies the fission sources guess to generate the next fission source, $\Omega_{ij}^{\ell+1} = A \Omega_{ij}^{\ell}$. Matrix A is obtained by rearrangement of the left side of Eq. (V-29).

Although the λ_i 's are generally unknown, an initial estimate, $\bar{\sigma}$, of the dominance ratio is obtained by observing the convergence rate of initial unaccelerated outer iterations (Refs. 23 and 25); that is,

$$\bar{\sigma} = \frac{\sum_{ij} |\Omega_{ij}^{\ell} - \Omega_{ij}^{\ell-1}|}{\sum_{ij} |\Omega_{ij}^{\ell-1} - \Omega_{ij}^{\ell-2}|} . \quad (V-91)$$

During the transient calculations, the flux iterations are started from an initial guess of the extrapolated flux from previous time steps. This guess is normally quite good because, by definition, the flux shape does not change significantly over a time step. Thus, for the transient calculations, the initial estimate of $\bar{\sigma}$ is chosen to be the larger of the previous time-step value or that given by Eq. (V-91). Later estimates (after the accelerated iterations have begun) are obtained by comparing the theoretical and actual convergence rates.

$$\bar{\sigma} = \frac{\sigma_0}{2} \cosh \left[\frac{\cosh^{-1}(\epsilon_\ell) C_\ell \left(\frac{2}{\sigma_0} - 1 \right)}{\ell} \right] + \frac{\sigma_0}{2} , \quad (V-92)$$

where C_ℓ denotes a Chebyshev polynomial of degree ℓ , and

$$\epsilon_\ell = \frac{\sum_{ij} |\Omega_{ij}^\ell - \Omega_{ij}^{\ell-1}|}{\sum_{ij} |\Omega_{ij}^\kappa - \Omega_{ij}^{\kappa-1}|} , \quad (V-93)$$

where κ indicates the outer iteration number where the current acceleration series was begun.

The constant dominance ratio, σ_0 , is used for each accelerated series of outer iterations. However, after each iteration, a new estimate, $\bar{\sigma}$, is determined, initially from Eq. (V-91) and for later iterations from Eq. (V-92). When a new series of outer iterations is accelerated, the constant dominance ratio, σ_0 , is set to the most recent value for $\bar{\sigma}$. If a dominance-ratio estimate is used that is too large, the fission source series will diverge, and the error estimate will grow. Such a situation can arise occasionally because of an unusual fuel spatial distribution or because of the use of previous time-step dominance-ratio estimates. An unaccelerated series (of length I_0) then is used to permit any high-order error modes to die away.

Because the rate of convergence is related directly to the accuracy of the dominance-ratio estimate, which is improved continually as the iteration proceeds, it is desirable to use a series of low- and intermediate-order acceleration polynomials instead of the single high-order polynomial C_ℓ . An accelerated sequence of outer iterations is terminated, and a new sequence is begun using an updated estimate for the eigenvalue dominance ratio whenever any of the following criteria are satisfied.

$$(1) \epsilon_\ell > 1.0 \text{ and } \frac{|\bar{\sigma} - \sigma_0|}{1 - \sigma_0} > 0.05 ,$$

$$(2) \ell - \kappa > I_{\max} , \text{ or}$$

$$(3) \bar{\sigma} \geq 1.0 ,$$

(V-94)

where $I_{\max}$ indicates the highest order Chebyshev polynomial permitted.

The initial unaccelerated iterations (used to obtain the initial dominance ratio estimate) are continued until all of the following conditions are met.

$$(1) \bar{\sigma} < 1.0 ,$$

$$(2) \bar{\sigma} > 0.4 ,$$

$$(3) \epsilon_{\lambda} > 0.2 ,$$

$$(4) \left| \frac{k_0^{\ell-1}}{k_0^{\ell}} - 1 \right| > 0.025 , \text{ and}$$

$$(5) \ell > I_0 ,$$

(V-95)

where I_0 indicates the minimum number of initial unaccelerated iterations permitted.

Note that the unaccelerated power method is used if I_0 is input as a large value.

The parameter ϵ_{λ} in Inequalities (V-95) is defined as

$$\epsilon_{\lambda} = \frac{\bar{\lambda} - \lambda}{2} ,$$

(V-96)

where

$$\bar{\lambda} = \max_{i,j} \left(\frac{\Omega_{ij}^{\ell}}{\Omega_{ij}^{\ell-1}} \right) \text{ and}$$

$$\underline{\lambda} = \min_{i,j} \left(\frac{\Omega_{ij}^{\ell}}{\Omega_{ij}^{\ell-1}} \right) .$$

The outer iterations are terminated when the following conditions are met for two successive iterations.

$$(1) \left| \frac{k_0^{\ell-1}}{k_0} - 1 \right| < \epsilon_0 \quad \text{and}$$

$$(2) \max \{ |1 - \bar{\lambda}|, 1 - \underline{\lambda}| \} < \epsilon_{pt} . \quad (V-97)$$

where ϵ_0 is the convergence precision specified by the user for the eigenvalue and ϵ_{pt} is the convergence precision for the pointwise fission source. If $\epsilon_{pt} = 0$, the second criterion is not used.

After the fission source is obtained, the total group source, S_{ijg} , for outer iteration ℓ is computed as

$$S_{ijg}^{\ell} = (1 - \beta) x_{pg} \Omega_{ij}^{\ell} + \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd} + \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'}^{\ell} , \quad (V-98)$$

where the steady-state pointwise precursor concentration is given by

$$C_{ijd}^{\ell} = \beta_d \frac{\Omega_{ij}^{\ell}}{\lambda_d} . \quad (V-99)$$

With the exception of the within-group scattering source, the right side of Eq. (V-29) is calculated, and the group flux calculation can begin. In summary, each outer iteration consists of the following steps.

- (1) Based on group fluxes from the previous outer iteration, previous time step, or initial flux guess and on the current fission cross sections, the fission source for the outer iteration is computed using Eq. (V-84).
- (2) The current fission source is compared with the fission source of the previous outer iteration to check convergence and to determine an eigenvalue estimate for the initial flux solution, Eq. (V-85).
- (3) If a real flux solution is to be obtained for a transient initial state, the adjustment factors are determined for the neutronics physical data. These factors are described in Appendix H.
- (4) The fission source is normalized by the current eigenvalue [Eq. (V-86)], and the integrated fission source error and the pointwise errors are compared with the criteria of Eqs. (V-97). If these errors satisfy the convergence criteria, the outer iteration is terminated.
- (5) If the conditions of Eq. (V-95) have been met, the fission source for the current iteration is determined by Chebyshev extrapolation [Eq. (V-87)]. Otherwise, the fission source is unaccelerated.
- (6) If the real initial flux solution is desired, the cell-dependent delayed-neutron precursor distribution is obtained by using Eq. (V-99).
- (7) The source to each flux group is obtained by using Eq. (V-98), and the new flux for each group is computed in the inner iteration as described in the next section.
- (8) Return to Step 1.

c. Adjoint Equation Solution. The SIMMER-II program solves both the adjoint transport equation and the adjoint diffusion equation by transposing the scattering and fission source matrices and inverting the group order of the problem. Transposition of the scattering matrix converts the normal downscattering problem to an upscattering problem. The downscattering dominance is restored by inverting the group order. The adjoint equation is solved before the real flux equation because the adjoint fluxes are required for the transient initial conditions (if used).

The solution of the adjoint transport equation differs from that of the diffusion solution only in that the solution for a given angular direction is identified as the adjoint flux solution for the opposite direction. However, this is of no practical consequence because only the scalar fluxes are required for further calculations.

2. Quasistatic Calculative Flow. The quasistatic algorithm requires three different calculations: (1) determination of the flux shape, (2) determination of reactivity and other amplitude equation parameters, and (3) solution of the amplitude equations. These three operations are characterized by different time scales and thus require three different time-step sizes. The flux shape is solved for time steps of the largest size, which are denoted by Δt^S . The amplitude equation parameters ρ , β_d , and Λ are evaluated for the time steps of intermediate size, called reactivity steps, and are denoted by Δt^R . The amplitude equation itself is solved at time steps of the smallest size, denoted by Δt^P . The general approach used to perform these calculations is illustrated in Fig. 19 and described in the following. The method is described in detail later.

First, new cross sections are calculated from the material densities and temperatures predicted by the fluid-dynamics calculation for the end of the previous reactivity step. These cross sections and updated fluxes are used to calculate new amplitude-equation parameters. The spatial distribution of specific material powers is updated in preparation for the next series of fluid-dynamics steps. The amplitude equation then is solved again for the current reactivity step. Next, the amplitude equation parameters are extrapolated quadratically to the end of the next reactivity step, and the amplitude equation is solved for the amplitude. The logarithm of this amplitude and the amplitude for the two previous reactivity steps is fitted to a parabola and used to calculate the power level for the subsequent fluid-dynamics calculation. This description summarizes the neutronics calculations that occur at the end of a reactivity step and prepare for the initiation of the next reactivity step. After the fluid-dynamics calculation for the next reactivity step is completed, the cycle continues until the end of the current flux-shape step is reached.

If the end of a shape step is reached, the cycle is interrupted after the calculation of the reactivity state resulting from the previous series of fluid-dynamic steps. Thus, a new flux shape is calculated after the new cross sections are determined. Amplitude-equation parameters resulting from the new flux shape are compared with the parameters from the extrapolated flux shape. Any change in the parameters is assumed to have accumulated linearly during the shape step. The amplitude-equation parameters are corrected for each reactivity step within the shape step, and the corresponding amplitude-equation solutions are recalculated. If the constraint condition is satisfied to within a given tolerance, the amplitude-equation parameters are extrapolated into the next

Fig. 19.
General approach used in the overall
quasistatic algorithm calculative flow.

reactivity step, and the calculation proceeds as from the beginning. Otherwise, the calculation proceeds as from the end of the shape step with the recalculation of the flux shape.

The correction of the amplitude-equation parameters for the difference between the initial quadratically extrapolated values and the final shape-step values results in a power discrepancy between the fluid-dynamics and neutronics calculations. To maintain a consistent total energy between the two calculations, the amplitude function is adjusted after its calculation from the amplitude equation to include the discrepancy in energy from the previous shape step. This discrepancy is generally a small fraction of the total energy calculated for the shape step.

a. Time-Step Controls. Time-step controls are used to anticipate the need to terminate a series of fluid-dynamic steps to update the amplitude function extrapolation (reactivity step) or to terminate a series of reactivity steps to update the flux shape (shape step). Generally, time-step controls monitor quantities that may affect the validity of the reactivity projection, such as a high reactivity derivative or a high reactivity state (implying a high power level). Other controls monitor the fluid-dynamics calculation to detect significant system changes, such as the maximum change in internal energy or density. Time-step controls for shape steps generally monitor quantities that reflect the rate of flux-shape changes, such as the rate of flux tilt from previous steps or the deviation of the quasistatic constraint at reactivity steps. Details are described for both types of time step controls in a later section.

The time-step controls for the quasistatic method regulate the length of both the reactivity step and the shape step. The tightness of the control depends on the reactivity of the system. The controls become very tight for a reactivity near prompt critical, but they are relaxed for a reactivity not near prompt critical. There are two time-step reduction factors with this reactivity dependency that appear in several of the individual time-step controls. The first reduction factor is given by

$$f_1(\rho) = 1 - (1 - \epsilon_4) * \text{EXP} \left[-\epsilon_{17} \left(\frac{\rho - \beta}{\beta} \right)^2 \right] , \quad (\text{V-100})$$

where ϵ_4 and ϵ_{17} are given by the user. The second reduction factor does not

tighten the time-step size as much as the first. Its form is fixed in SIMMER-II as

$$f_2(\rho) = f_1(\rho)^{1/5} . \quad (V-101)$$

A plot of these two reduction factors as a function of reactivity in dollars is shown in Fig. 20. The input parameters ϵ_4 and ϵ_{17} used to create these curves have the suggested values of 0.02 and 0.08515781.

The first reduction factor is identical to the one used in QX1 if $\epsilon_{17} = 100 \ln 2$. This value of the parameter only tightens the time-step control as the reactivity nears prompt critical. However, the SIMMER-II recommended values tighten the time-step control as the system nears criticality and tightens them even more as the system nears prompt critical. In preparing the time-step control parameters, the user should recall that $f_1 = 0.1$ and $f_2 = 0.63$ at $\rho = 0$ for the recommended values of ϵ_4 and ϵ_{17} . The input values of the time-step controls for $\rho \rightarrow \infty$ may have to be large to avoid time-step controls that are unnecessarily tight.

During the transient calculation, the shape time step is predicted or limited by seven separate controls over reactor or problem parameters. The individual controls are as follows.

- (1) Previous step size. The shape step is limited to 10 times the maximum of the previous two shape time steps.
- (2) Change in leakage. The shape time step depends on the rate of change in the leakage contribution to the reactivity according to

$$\Delta t^{s2} = f_1(\rho) \epsilon_6 \left| \frac{\Delta t^s I^r + (t^r - t^s) I^{s-1}}{I^r - I^{s-1}} \right| , \quad (V-102)$$

where

- ϵ_6 denotes the time-step control specified by the user limiting the relative change in the leakage contribution to the reactivity,
- I^r denotes the extrapolated leakage contribution to the reactivity for the most recent reactivity step,
- t^r denotes the time for the most recent reactivity step,

Fig. 20.
Functions used for time-step control of the quasistatic method algorithm.

t^s denotes the time for the most recent shape step, and

$$I^{s-1} = -\sum_{i,j} \sum_{g=1}^{IGM} \phi_{ijg}^* [FR_{i+1/2j}^{s-1} - FR_{i-1/2j}^{s-1} - FL_{i+1/2j}^{s-1} + FL_{i-1/2j}^{s-1} + FU_{ij+1/2j}^{s-1} - FU_{ij-1/2}^{s-1} - FD_{ij+1/2}^{s-1} + FD_{ij-1/2}^{s-1}] , \quad (V-103)$$

denotes the leakage contribution to the reactivity for the previous shape calculation. [Note: The terms used in Eq. (V-103) are defined in Appendix L.]

- (3) Number of reactivity steps. The shape time step cannot extend over more than a number of reactivity steps specified by the user.
- (4) Problem time. A shape calculation is taken at the end of the problem.
- (5) Maximum shape step. The shape step is limited by a maximum size specified by the user.
- (6) Change in shape. The tilt of the total flux shape is calculated as the relative change in the flux at each mesh point:

$$T_{ij} = \frac{\sum_{g=1}^{IGM} \phi_{ijg}^s - \sum_{g=1}^{IGM} \phi_{ijg}^{s-1}}{\sum_{g=1}^{IGM} \phi_{ijg}^s} , \quad (V-104)$$

where

ϕ_{ijg}^s denotes the most recently calculated flux and

ϕ_{ijg}^{s-1} denotes the previous calculated flux shape.

By comparing the values of the flux tilt throughout the mesh, the shift of the flux from one region of the mesh to another can be monitored and used to estimate the shape time step. This time step is determined from

$$\Delta t^{s8} = \frac{f_1(\rho) \Delta t^s \epsilon_9}{\max_{ij} |T_{ij}| - \min_{ij} |T_{ij}|} , \quad (V-105)$$

where

Δt^s denotes the shape time step between the time t^s and t^{s-1} and

ϵ_9 denotes the time-step control specified by the user for monitoring the flux tilt;

- (7) Change in reactivity. The shape step is predicted by the change in reactivity:

$$\Delta t^{s9} = t^r - t^s + \max[0.0, \frac{f_1(\rho) \epsilon_{18} \beta^r - |\rho^r - \rho^s|}{\Lambda^r |a_1 + a_2 \Delta t^r|}] , \quad (V-106)$$

where

ϵ_{18} denotes the time-step control specified by the user for the maximum reactivity change (in dollars) per shape step,

β^r is the effective delayed neutron fraction for the most recent reactivity step,

ρ^r is the reactivity for the most recent reactivity step,

ρ^s is the reactivity for the most recent shape calculation,

Λ^r is the neutron generation time for the most recent reactivity step, and

a_1 and a_2 are the first- and second-order coefficients of the parabolic fit to ρ/Λ for the most recent reactivity time step. The final shape step is calculated from the minimum of these seven estimated time steps.

During the transient neutronics calculation, the reactivity time step is predicted or limited by six separate controls over reactor or problem parameters. The individual controls are as follows.

- (1) Previous step size. The reactivity step is limited to twice the maximum of the previous two reactivity steps.

- (2) Change in reactivity. The reactivity step is predicted by the maximum reactivity change permitted.

$$\Delta t^{r2} = \frac{f_1(\rho) \epsilon_5 \beta^r}{\Lambda^r |a_1 + a_2 \Delta t^r|} , \quad (V-107)$$

where ϵ_5 denotes the time-step control specified by the user for limiting the maximum reactivity change per reactivity step.

- (3) Shape-step size. The reactivity step is limited by the shape.
 (4) Maximum step size. The reactivity step is limited to a maximum size specified by the user.
 (5) Change in amplitude. The reactivity time step is limited by the rate of change in the flux amplitude solution by solving

$$\ln \left(\frac{N_0 + N_1 \Delta t^{r3} + N_2 \Delta t^{r3}{}^2}{N^r} \right) = \frac{2.3 f_2(\rho)}{\epsilon_7} , \quad (V-108)$$

for Δt^{r3} , where ϵ_7 denotes the time-step control specified by the user that determines the number of reactivity steps per power decade.

- (6) Maximum inverse period. The reactivity time step is limited by the change in inverse period,

$$|\omega^r - \omega^s| \leq \epsilon_i f_2(\rho) , \quad (V-109)$$

where ϵ_8 denotes the time-step control specified by the user for the maximum inverse period change per reactivity step. The reactivity time step is given by the minimum of these six time steps.

A reactivity step also may be forced by changes that occur during the fluid-dynamics calculation. The limits are applied to changes in amplitude, material components, and component energies. The first of these limits controls the number of reactivity steps per power decade according to

$$|\Delta t^h b_1 + (\Delta t^h)^2 b_2| \leq \frac{f_2(\rho)}{2.3 \epsilon_{10}} , \quad (V-110)$$

where

Δt^h denotes the time interval from the start of the most recent reactivity step to the end of the most recent fluid-dynamics step,

b_1 and b_2 are the first- and second-order coefficients in the exponential fit to the amplitude function over the most recent reactivity step, and

ϵ_{10} denotes the control specified by the user that determines the number of reactivity steps per amplitude decade.

If this inequality is violated, the fluid-dynamics calculation is suspended and control is returned to the neutronics calculation to perform the next reactivity step. In that case, the reactivity time-step size is set to Δt^h .

The second control limits the change in component density per reactivity step. For the fuel density, this control is given by

$$\sum_{\eta=n^r}^{n-1} \left[\frac{\sum_{i,j} |(\bar{\rho}_{fuel})_{ij}^{\eta+1} - (\bar{\rho}_{fuel})_{ij}^{\eta}| v_{ij}}{\sum_{i,j} (\bar{\rho}_{fuel})_{ij}^{\eta+1} v_{ij}} \right] < f_1(\rho) \epsilon_{14} , \quad (V-111)$$

where

ϵ_{14} denotes the control specified by the user that limits the maximum absolute fuel mass shift in a reactivity step,

n^r is the number of the fluid-dynamics time step that began the current reactivity step, and

n is the most recent fluid-dynamics time step number.

The fuel density is calculated by summing over all fuel solid and liquid components.

$$\bar{\rho}_{fuel} = \bar{\rho}_{s1} + \bar{\rho}_{s2} + \bar{\rho}_{s3} + \bar{\rho}_{s4} + \bar{\rho}_{L1} + \bar{\rho}_{L2} + \bar{\rho}_{L6} + \bar{\rho}_{L7} . \quad (V-112)$$

If the inequality is not satisfied after a fluid-dynamics time step, the fluid-

dynamics calculation is suspended and a reactivity step is taken. Similar controls apply to the steel and sodium mass changes and are controlled by the input quantities, ϵ_{15} and ϵ_{16} , respectively.

The third control limits the change in material energy. For the fuel energy, this control is given by

$$\sum_{\eta=n}^{n-1} \left[\frac{\sum_{i,j} |(\bar{\rho}_{fuel} e_{fuel})_{ij}^{\eta+1} - (\rho_{fuel} e_{fuel})_{ij}^{\eta}| v_{ij}}{\sum_{i,j} (\bar{\rho}_{fuel} e_{fuel})_{ij}^{\eta+1} v_{ij}} \right] < f_1(\rho) \epsilon_{11} \quad (V-113)$$

The total fuel energy is obtained by summing over all fuel components.

$$(\bar{\rho}_{fuel} e_{fuel}) = \bar{\rho}_{S1} e_{S1} + \bar{\rho}_{S2} e_{S2} + \bar{\rho}_{L1} e_{L1} + \bar{\rho}_{L5} e_{L5} \quad (V-114)$$

The time-step control factor that limits the maximum fuel energy change per reactivity step, ϵ_{11} , is input by the user. Similar limits apply to the steel and sodium energy change and are controlled by the input quantities ϵ_{12} and ϵ_{13} , respectively.

b. Initialization. The specific power shape for each material in each mesh cell is calculated as

$$Q_{ij}^m = \frac{1}{\rho_{ij}^m} \sum_{g=1}^{IGM} (\alpha_c^m \Sigma_{c,ijg}^m + \alpha_f^m \Sigma_{f,ijg}^m) \phi_{ijg} \quad (V-115)$$

where

ρ_{ij}^m denotes the microscopic density for material m,

α_c^m denotes the energy conversion factor per neutron captured in material m (may be group-dependent if NEUPREP is on),

$\Sigma_{c,ijg}^m$ denotes the macroscopic capture cross section in energy group g for material m of density ρ_{ij}^m ,

α_f^m denotes the energy conversion factor per fission in material m (may be group-dependent if NEUPREP is on), and

$\Sigma_{f,ijg}^m$ denotes the macroscopic fission cross section in energy group g for material m of density ρ_{ij}^m .

The total power in the reactor system is calculated as

$$P = N(0) \sum_{i,j} V_{ij} \sum_{m=1}^{NC} Q_{ij}^m \bar{\rho}_{ij}^m, \quad (V-116)$$

where

$N(0)$ denotes the initial value of the amplitude function ($\approx$ unity)

$\bar{\rho}_{ij}^m$ denotes the macroscopic density of material m in the mesh cell (i,j) , and

NC denotes the total number of neutronics components (five in SIMMER-II: fertile fuel, fissile fuel, steel, sodium, and control material).

The initial flux solution is unnormalized. To obtain the desired initial power, the fluxes are multiplied by the ratio of the desired power to the calculated power. The initial precursor concentrations and the material-specific power also are normalized by this factor. The constraint constant in Eq. (V-28) normalizes the adjoint flux.

c. Reactivity Evaluation Steps. The details of the reactivity steps are quite different, depending on the choice of flux-shape extrapolation or flux-shape update as specified by the user. If flux-shape extrapolation is selected, the flux shape used to recalculate the reactivity after a series of fluid-dynamic steps is extrapolated from the two previous flux-shape solutions at shape steps. This approach requires much less computational effort; however, success depends on the occurrence of essentially linear changes in flux shape. The effect of threshold or nonlinear changes in fluid-dynamic behavior will not be recognized until the end of the shape step when the flux shape is recalculated. If significant changes in reactivity result from the flux-shape change, the iteration that occurs over the series of reactivity steps and at the end of shape steps will not produce accurate results. (See Ref. 26 for a discussion.) The second approach is based on an update of the flux shape by the flux-shape solution module at each reactivity step. In essence, an abbreviated series of outer iterations is performed to improve the flux shape but not to provide a fully converged solution. Thus, the effect of changes resulting from fluid-dynamic effects appears in the flux shape at each reactivity step. Figure 21 shows the calculative flow for the quasistatic algorithm if flux-shape extra-

polaion is used at reactivity steps, and Fig. 22 shows the calculative flow if flux-shape updating is used. Figure 23 is a detailed flow-chart of the 7-iteration portion of the quasistatic method algorithm for both flux-shape options.

In both approaches, the first step is to calculate new cell-dependent cross sections corresponding to the new configuration (that is, new cell material densities and temperatures). If the shape extrapolation option is used, the flux shape is extrapolated to the end of the current reactivity step based on flux shapes calculated at the two previous shape steps. Next, the amplitude equation coefficients are calculated from the current cross sections and the extrapolated flux shapes or, if the flux-shape update option is used, the flux shape from the previous reactivity step. In both methods, the amplitude equation then is solved for the current reactivity step based on the updated coefficients. The spatial delayed-neutron source is updated based on the current and previous reactivity-step fission sources and the integrating factors determined in the amplitude-equation solution. Checking for the end of the current shape step is the next step, which completes the sequence for the shape extrapolation option.

In the flux-shape update option, the calculation is continued by the flux-shape calculation module, with the flux shape from the previous reactivity step provided as an initial guess. Convergence criteria are provided that are less restrictive than the criteria used at shape steps. For small changes in the reactor configuration, usually one outer iteration will meet the convergence criteria. More unusually, significant changes may require three or four outer iterations. After the flux shape is converged, the amplitude-equation coefficients are recalculated based on the new flux shape. The spatial material power distribution also is recalculated. The amplitude equation again is solved for the current reactivity step. After recalculation of the time-step sizes, a check is made to determine if the end of the current shape step has been reached.

If the current shape step has not been completed, the amplitude equation parameters are projected in both the extrapolation and the update method. The projection is parabolic and based on the current and two previous reactivity step values. The amplitude equation then is solved for the next reactivity step. The amplitude variation over the next reactivity step is characterized by fitting a parabola in log amplitude to the amplitude at the end of the next reactivity step, the current amplitude, and the amplitude at the previous reactivity step.

Fig. 21.
Calculative flow of the quasistatic method
algorithm for the extrapolated flux-shape option.

Fig. 22.
Calculative flow of the quasistatic method
algorithm for the updated flux-shape option.

Fig. 23.
Calculative flow of the 7-iteration portion of the
quasistatic method algorithm for both flux-shape options.

The calculation then is transferred to the fluid-dynamics modules. A series of fluid-dynamics steps is completed, and the reactivity-step cycle begins again.

d. Flux-Shape Evaluation Steps. In both the flux-shape extrapolation and the flux-shape update method, a check for the end of the current shape step is done at the end of each reactivity step. If a shape step is needed, the following procedure is followed.

A fully converged flux shape is calculated. The amplitude-equation coefficients then are calculated based on the new flux shape. If the flux-shape-step iteration is converged, which is known at this point because convergence depends on the quasistatic constraint integral, which in turn was determined during the flux-shape calculation, the spatial material power distribution is calculated for the next reactivity step. Next, the changes in the amplitude-equation coefficients, from the previous iteration or the first iterate values at the end of the current shape step, are distributed linearly over all the reactivity steps composing the current shape step. The series of amplitude-equation solutions for each reactivity step then is repeated. The current value of the amplitude function is available for the next iterative solution of the flux-shape equation or extrapolation of the power for the next series of fluid-dynamics steps. During this procedure, the spatial delayed-neutron-precursor concentrations are integrated, and the value is compared with the effective precursor concentrations used in the amplitude equations. If a significant difference is detected, the spatial concentrations are modified to agree with the amplitude-equation values because the numerical method used in the amplitude equation solution is considered to be more accurate and because consistency is necessary if the quasistatic constraint integral is to converge to unity.

This iterative process is completed when the quasistatic constraint condition is satisfied to within a specified tolerance, until the change in γ between two consecutive iterations is less than a tenth of the tolerance, or until the number of iterations exceeds a specified maximum. Both the tolerance and the maximum number of iterations are provided by the user.

D. Point Reactor Model (URANUS) Option

The point reactor model is included as an inexpensive neutronics option in SIMMER-II to permit methods and model development and to allow exploratory calculations in which space-dependent neutronics effects are not considered. The

point reactor model option allows the use of component reactivity worths for whole-core reactor problems or a specified power vs time for single subassembly problems. The basic structure of the SIMMER-II point reactor model is taken from the SAS3A⁶ code and retains its decay heat model.

1. Point Reactor Model Equations. For the point reactor model, neutron flux separability is assumed as in the quasistatic method, but the neutron flux shape is treated as being time-independent. The power shape and the material reactivity worths are assumed also to be time-independent, and the convection and phase change of delayed neutron precursors are neglected. The point reactor model equations [Eqs. (III-260) and (III-261)] are solved using the same technique²⁷ as is applied to the quasistatic amplitude equation.

The point reactor model contains a decay heat model to describe the buildup and decay of fission products that also contribute to power generation. The incorporation of this model into the point reactor model is described in Ref. 6.

The reactivity used in [Eq. (III-260)] consists of five components.

$$\rho = \delta\rho_p + \delta\rho_{Dop} + \delta\rho_{Fert} + \delta\rho_{Fiss} + \delta\rho_{SS} + \delta\rho_{Na} + \delta\rho_{Cont} \quad , \quad (V-117)$$

where

$\delta\rho_p$ denotes the driving or programmed reactivity,

$\delta\rho_{Dop}$ denotes the Doppler or temperature reactivity feedback,

$\delta\rho_{Fert}$ denotes the fertile fuel displacement reactivity feedback,

$\delta\rho_{Fiss}$ denotes the fissile fuel displacement reactivity feedback,

$\delta\rho_{SS}$ denotes the stainless-steel displacement reactivity feedback,

$\delta\rho_{Na}$ denotes the sodium displacement reactivity feedback, and

$\delta\rho_{Cont}$ denotes the control material displacement reactivity feedback.

The programmed reactivity, $\delta\rho_p$, permits users to specify a reactivity change not predicted by material motion calculated in the fluid dynamics. The Doppler reactivity feedback is assumed to result only from the heating of the fertile fuel component and is calculated according to

$$\delta\rho_{\text{Dop}} = \sum_{i,j} (\bar{\rho}_{\text{Fert}})_{ij} V_{ij} (DW)_{ij} \ln \left[\frac{(\bar{T}_{\text{Fert}})_{ij}}{(\bar{T}_{\text{Fert}}^0)_{ij}} \right] , \quad (\text{V-118})$$

where

$(\bar{T}_{\text{Fert}}^0)_{ij}$ is the initial mass average temperature of the fertile fuel in cell (i,j) and

$(DW)_{ij}$ denotes the Doppler weight for cell (i,j).

The weights DW_{ij} are provided by the user and initially are normalized such that

$$\sum_{i,j} (\bar{\rho}_{\text{Fert}})_{ij} V_{ij} (DW)_{ij} = (DC) = T \frac{dk}{dT} , \quad (\text{V-119})$$

where the Doppler coefficient, (DC), is provided by the user.

The displacement feedback reactivities are calculated in a similar manner. For example, the fertile fuel displacement reactivity feedback is given by

$$\delta\rho_{\text{Fert}} = \sum_{i,j} [(\bar{\rho}_{\text{Fert}})_{ij} - (\bar{\rho}_{\text{Fert}}^0)_{ij}] V_{ij} (W_{\text{Fert}})_{ij} , \quad (\text{V-120})$$

where

$(\bar{\rho}_{\text{Fert}}^0)_{ij}$ denotes the initial fertile fuel macroscopic density in mesh cell (i,j) and

$(W_{\text{Fert}})_{ij}$ denotes the fertile fuel reactivity worth in $\delta k/kg$ for cell (i,j).

The reactivity worths and Doppler weights are assumed to be separable in space to limit the amount of input, for example,

$$W_{\text{Fert}}(r,z) = W_{\text{Fert}r}(r) W_{\text{Fert}z}(z) . \quad (\text{V-121})$$

For the reactivity worths to have the appropriate units, one of the vectors $W_{\text{Fert}r}$ or $W_{\text{Fert}z}$ should have the correct units, $\Delta k/kg$, and the other should be normalized to give a maximum of one. If the separability assumption is

considered inappropriate for some calculations, the SIMMER-II program can be modified to use pointwise worths and weights.

2. Energy Source. The point-kinetics energy source distribution, Q_{ij} , can be input with units of power per unit mass or power per unit volume. The latter option is convenient when the power density distribution is obtained from separate neutronics analyses.

SIMMER-II internally normalizes Q_{ij} for the power-per-unit-mass option by requiring

$$\sum_{i,j} Q_{ij} V_{ij} [(\bar{\rho}_{\text{Fert}}^0)_{ij} \gamma_{\text{Fert}} + (\bar{\rho}_{\text{Fiss}}^0)_{ij} \gamma_{\text{Fiss}} + (\bar{\rho}_{\text{SS}}^0)_{ij} \gamma_{\text{SS}} + (\bar{\rho}_{\text{Na}}^0)_{ij} \gamma_{\text{Na}} + (\bar{\rho}_{\text{Cont}}^0)_{ij} \gamma_{\text{Cont}}] = P_0 \quad (\text{V-122})$$

or, for the power-per-unit-volume option, by requiring

$$\sum_{i,j} Q_{ij} V_{ij} = P_0, \quad (\text{V-123})$$

where

γ_m is the fraction of power generated in component m ;

$\bar{\rho}_m^0$ is the initial total macroscopic density of component m in mesh cell ij , where m represents Fert, Fiss, SS, Na, and Cont; and

P_0 is the initial total power in the region of solution.

The specific (power-per-unit-mass) energy source Q_{mij} for component m in mesh cell ij then is defined to be

$$Q_{mij} = \gamma_m Q_{ij} \quad (\text{V-124})$$

for the power-per-unit-mass option or to be

$$Q_{mij} = \frac{\gamma_m Q_{ij}}{[\bar{\rho}_{\text{Fert}}^0 \gamma_{\text{Fert}} + \bar{\rho}_{\text{Fiss}}^0 \gamma_{\text{Fiss}} + \bar{\rho}_{\text{SS}}^0 \gamma_{\text{SS}} + \bar{\rho}_{\text{Na}}^0 \gamma_{\text{Na}} + \bar{\rho}_{\text{Cont}}^0 \gamma_{\text{Cont}}]} \quad (\text{V-125})$$

for the power-per-unit-volume option.

Units of power per unit volume generally are more convenient for Q_{ij} . However, this case can be indeterminate when $[\beta_{Fert}^0 \gamma_{Fert} + \beta_{Fiss}^0 \gamma_{Fiss} + \beta_{SS}^0 \gamma_{SS} + \beta_{Na}^0 \gamma_{Na} + \beta_{Cont}^0 \gamma_{Cont}]$ is zero. Even a proper calculation of Q_{ij} would give zero, and Q_{mij} would be indefinite. In this situation, SIMMER-II sets Q_{mij} equal to zero internally. As a result, no power will be generated in mesh cell ij even though fission and gamma-heating material may convect into the mesh cell at a later time. This can be avoided with the power-per-unit-mass option by giving a finite value for Q_{ij} to material not initially present. Then when such material is convected into the mesh cell, power will be generated in the mesh cell. Even though this seems more realistic, the accuracy of the point kinetics model in SIMMER-II becomes questionable after significant material relocation has occurred.

3. Point Reactor Model Calculative Flow. The feedback reactivities are available only after a fluid-dynamics time step has been completed, but the energy source is needed during the calculation for each time step. To obtain the energy source for the next time step, the total reactivity for the previous time step is extrapolated, and the point kinetics equations are solved for the current time step. Because this extrapolation could be in error when the feedback reactivities are obtained correctly for the current time step, the point kinetics equations are solved again for the time step, and any energy discrepancy is included in the next time step.

4. Point Reactor Model Time-Step Controls. In addition to the normal fluid-dynamics time-step controls described in Sec. IV.G, two others are imposed on the fluid-dynamics time step by the point reactor model. The first control, DTL(9), limits the maximum power change that can occur in a time step by requiring

$$\Delta t \leq \frac{\delta_p N^{n-1} \Delta t_p}{|N^n - N^{n-1}|} \quad , \quad (V-126)$$

where

Δt is the fluid-dynamics time step,

ϵ_p denotes the maximum fractional power change desired in a

time step, and

Δt_p is the previous fluid-dynamics time step.

The second control, DTL(10), limits the maximum reactivity change in a time step and is similar to that used in the quasistatic method. The time step is limited such that

$$\Delta t \leq \epsilon_\rho \left\{ 1 - 0.9 \exp \left[-100 \ln 2 \left(\frac{\rho - \beta}{\beta} \right)^2 \right] \right\} \beta \frac{\Delta t_p}{|\rho^n - \rho^{n-1}|}, \quad (V-127)$$

where the expression in brackets is the same as Eq. (V-100), where $\epsilon_4 = 0.1$ and $\epsilon_{17} = 100 \ln 2$. This expression is the one used in QX1³ and tightens the time step control by a factor of 10 at $\rho = \beta$ and to 0.55 ϵ_ρ at $|\rho - \beta| = 0.1\beta$.

REFERENCES

1. K. D. Lathrop and F. W. Brinkley, "TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport," Los Alamos Scientific Laboratory report LA-4848-MS (July 1973).
2. K. L. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems," Argonne National Laboratory report ANL-82-64 (1983).
3. D. A. Meneley, K. O. Ott, and E. S. Weiner, "Fast-Reactor Kinetics -- The QX1 Code," Argonne National Laboratory report ANL-7769 (March 1971).
4. R. A. Shober, T. A. Daly, D. R. Ferguson, "FX2-TH: A Two-Dimensional Nuclear Reactor Kinetics Code with Thermal-Hydraulic Feedback," Argonne National Laboratory report ANL-78-97 (October 1978).
5. J. E. Cahalan and D. R. Ferguson, "A Preliminary User's Guide to Version 1.0 of the SAS3D LMFBR Accident Analysis Computer Code," Argonne National Laboratory unnumbered report (July 1977).
6. F. E. Dunn, W. R. Bohl, T. J. Heames, G. Hoppner, M. G. Stevenson, L. L. Smith, J. R. Travis, W. Woodruff, L. W. Person, J. M. Kyser, and D. R. Ferguson, "The SAS3A LMFBR Accident Analysis Computer Code," Argonne National Laboratory report ANL/RAS 75-17 (April 1975).
7. L. P. Abagyan, N. O. Bazazyants, I. I. Bondarenko, and M. N. Nikolaev, Group Constants for Nuclear Reactor Calculations (Consultants Bureau Enterprises, Inc., New York, 1964).

8. F. Parker, M. Ishikawa, and L. Luck, "MXS Cross-Section Preprocessor User's Manual," Los Alamos National Laboratory report LA-10856-M, NUREG/CR-4765 (March 1987).
9. M. Ishikawa and L. B. Luck, "An Improved Method for Cross-Section Evaluation for Reactor Safety Calculations," Power Reactor and Nuclear Fuel Development Corporation draft report (in preparation).
10. L. L. Smith and P. D. Soran, "Bondarenko Formalism for Multi-Isotope Materials" American Nuclear Society 1976 annual meeting Toronto, Ontario, Canada, (June 13-18, 1976).
11. B. M. Carmichael, "Standard Interface Files and Procedures for Reactor Physics Codes," Los Alamos Scientific Laboratory report LA-5436-MS (February 1974), Version III.
12. H. Takano and Y. Matsui, "Accuracy of Interpolation Methods for Resonance Self-Shielding Factors," J. Nuc. Sci. and Tech. 18(2), 152-161 (February 1981).
13. E. Kiefhaber and K. O. Ott, "Interpolationsformel für Selbstabschirmungsfaktoren," Kernforschungszentrum, Karlsruhe, report KfK 384 (1965).
14. M. Segev, "A Theory of Resonance - Group Self-Shielding" Nuc. Sci. and Engr. 50, 72-82 (1975).
15. M. Segev, "A Many-Resonance Approximation for the Neutron Energy Spectrum," Nuc. Sci. and Engr. 50, 354-363 (1973).
16. D. Woll, "GRUCAL, Ein Programm System zur Berechnung makroskopischer Gruppen Konstanten," Kernforschungszentrum, Karlsruhe, report KfK 2108 (June 1975).
17. E. Kiefhaber, "Anforderungen an ein verbessertes Verfahren zur ϕ_0 -Interpolation von f-Faktoren," Kernforschungszentrum, Karlsruhe, report INR 656 (September 1976).
18. F. H. Fröhner, "Self-shielding Factors Greater than Unity in Narrow-Resonance Approximation," Kernforschungszentrum, Karlsruhe, report INR 779 (March 1978).
19. C. de Boor, A Practical Guide to Splines (New York, Springer-Verlag, (1978).
20. K. O. Ott and D. A. Meneley, "Accuracy of the Quasistatic Treatment of Spatial Reactor Kinetics," Nuc. Sci. and Engr. 36, 402 (1969).
21. K. D. Lathrop, "Spatial Differencing of the Transport Equation: Positivity vs Accuracy," J. Comp. Phys. 4, 475 (1969).
22. W. H. Reed, "The Effectiveness of Acceleration Techniques for Iterative Methods in Transport Theory," Nuc. Sci. and Engr. 45, 245-254 (1971).
23. L. A. Hageman, "Numerical Methods and Techniques Used in the Two-dimensional Neutron-Diffusion Program PDQ-5," Bettis Atomic Power Laboratory report WAPD-TM-364 (February 1963).

24. R. S. Varga, Matrix Iterative Analysis, (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1962), p. 57.
25. A. Hageman, "The Estimation of Acceleration Parameters for the Chebyshev Polynomial and the Successive Overrelaxation Iteration Methods," Bettis Atomic Power Laboratory report WAPD-TM-1038 (June 1972).
26. L. B. Luck, "An Improved Quasistatic Method for Transition-Phase Calculations," Transactions of the American Nuclear Society, 41, 317, (June 1982).
27. J. J. Kaganove, "Numerical Solution of the One-Group Space-Independent Reactor Kinetics Equation for Neutron Density Given the Excess Reactivity," Argonne National Laboratory report ANL-6132 (1960).

VI. PROGRAM DESCRIPTION

The techniques and other information required to use and modify SIMMER-II are described in this chapter. The first section describes the steps required to create a compiler source of the SIMMER-II program. The programming techniques and data management are described in the next two sections (Secs. VI.B and VI.C). Input and output are described in the next three sections (Secs. VI.D-VI.F). Finally, restarting an interrupted calculation, postprocessing results, and use of system-dependent features are treated.

A. Overall Program Structure

In contrast to the usual method of selecting code options through input parameters for the executable code, the major options in the SIMMER-II program are selected by a maintenance processor^{1,2} according to user-specified instructions. The processor creates a FORTRAN source file with the desired options that then is compiled in the usual way. Using a maintenance processor reduces redundant programming, storage requirements, and execution times. However, it is not as flexible as having all options in a single-load module and requires a recompilation and reloading whenever a new option is desired. To avoid this drawback, several different load modules having the desired options can be stored.

1. Creating the FORTRAN Compiler Source. The CDC UPDATE¹ or OPCODE HISTORIAN² programs are used to maintain a source file. In SIMMER-II development, UPDATE has been used for inserting and deleting lines of FORTRAN coding, selecting or rejecting specified FORTRAN lines in the source for compilation, retrieving sets of FORTRAN lines (for example, common blocks) and inserting them in appropriate subroutines, and retrieving selected routines for compilation. These latter capabilities are required to process the SIMMER-II source program before it can be compiled.

Two types of executions are done with the processor. The first is a creation run, which is required to create a program library from the SIMMER-II source provided to each installation. The SIMMER-II source is used as input and, from the occurrence of *DECK* or *COMDECK* as the first card in the input stream, the processor automatically determines that a program library is being

*The meaning of these directions or preprocessor instructions is explained in the manuals for UPDATE or HISTORIAN.

created. The creation run creates a directory of DECKs and COMDECKs, thus eliminating the need for frequent rewinding of the old program library. Correction files that add new DECKs or COMDECKs should use the ADDFILE* option.

The other type of execution is a correction run that can be used to generate a new program library, a compiler source, or both from an old program library and, optionally, add correction sets. The order of DECKs and COMDECKs in a program library is important for correction runs. Therefore, the user should note the order in a creation run and as modified by ADDFILE directives in subsequent correction runs. After a correction run, a FORTRAN version of SIMMER-II is produced that is ready for compilation and loading.

a. SIMMER-II Source File. The SIMMER-II source file supplied to the user has been constructed at Los Alamos for use with the CDC UPDATE program or OPCODE HISTORIAN. The processor directives, *IF, *CALL, *DECK, *COMDECK, and *ENDIF, have been used extensively in the source file. The order of the decks in the SIMMER-II source file is

- (1) common decks,
- (2) program decks,
- (3) subroutine, and
- (4) function decks.

The common decks contain all FORTRAN common, dimension, equivalence, and LCM statements that are common among the program, subroutines, and functions. The program decks contain all programs necessary to construct the several overlays. The subroutine and function decks contain all the subroutines and functions necessary for the SIMMER-II program that are not contained in the system user's library. These routines are intended to be placed in a user's library.

To create a set of binary files for execution on a system that will not force-load the main program, two compilations must be made. First, all programs necessary for the overlays are extracted from the source file and are compiled. Then all subroutines and functions are extracted and compiled. A user's library is generated using the binary files from the second compilation, and the binary file containing the programs is loaded with the user's library as the primary library and the systems library as the secondary library. The resulting binary file is ready for execution.

*The meaning of these directions or preprocessor instructions is explained in the manuals for UPDATE or HISTORIAN.

b. Description of the SIMMER-II Code Options. The processor directive *IF allows many options in the SIMMER-II source file and allows several complete programs to be contained in one source file. The options in the present SIMMER-II source file are listed below in two categories. The first contains those options relating to the different operating systems and compilers. The second category contains the different calculative models.

(1) Operating System and Compiler Options.

(1) CRAY-on	Create a SIMMER-II program for the Cray-1 computer.
-off	Do not create a SIMMER-II program for the Cray-1 computer.
(2) CFT-on	Select the Cray FORTRAN compiler. FTNLTS must be on for the Los Alamos CTSS system.
-off	Do not select the Cray FORTRAN compiler.
(3) SCOPE-on	Create a SIMMER-II program for the CDC SCOPE system.
-off	Do not create a SIMMER-II program for the CDC scope system.
(4) FTN42-on	Create a SIMMER-II program for the CDC FTN compiler.
-off	Do not create a SIMMER-II program for the FTN compiler.
(5) FTNLTS-on	Create a SIMMER-II source file for the LTSS system compiler. FTN42 option must be on for the LTSS version. Do not turn on FTN42 if CFT is on.
-off	Do not create a SIMMER-II source file for the LTSS system compiler.
(6) IBM-on	Create a SIMMER-II program for IBM computers and systems.
-off	Do not create a SIMMER-II program for IBM computers.
(7) DEC-on	Create a SIMMER-II source file for the DEC FORTRAN compiler or the IBM G-FORTRAN compiler. The IBM option must be on.
-off	Do not create a source file for the DEC or the IBM G-FORTRAN compiler (use the IBM H-FORTRAN compiler).
(8) VAX-on	Create a SIMMER-II program for the VAX/VMS operating system.
-off	Do not create a SIMMER-II program for the VAX/VMS operating system.

- (9) DBL-on Create a SIMMER-II code that uses double-precision functions and constants.
- off Use single precision.
- (10) OVLAY-on Complete use of overlays.
- off Delete all use of overlays.
- (11) NOLCM-on Machine has only one core memory.
- off Machine has two memories, a SCM and a LCM.
- NOTE: NOLCM must be off for the CDC 7600 machines.
- (12) LCM-on Use the LCM array AA1 for the fluid-dynamics mesh cell data. LMEM must be off.
- off Use the SCM array AASC for all fluid-dynamics mesh cell data. If LCM is off, the problem is limited to $NQ*(IBAR+2)*(JBAR+2) \leq 5356$ for the CDC 7600.
- (13) LMEM-on Equivalence the array AASC to the LCM array AA1.
- off Use SCM for the array AASC.
- NOTE: If LMEM is on, LCM must be off.
- (14) NOSHUF-on Do not shuffle data memory between calls to the fluid-dynamics and neutronics sections of the code.
- off Shuffle data memory between calls to the fluid-dynamics and neutronics sections of the code.
- NOTE: If NOSHUF is on, NOLCM and LMEM must be on and LCM must be off. The 5356-word limitation on the mesh does not apply.
- (15) FOURB-on A eight-bit/byte four-byte/word machine.
- off Either EIGHTB or TENB is on.
- (16) EIGHTB-on A eight-bit/byte eight-byte/word machine.
- off Either FOURB or TENB is on.
- (17) TENB-on A six-bit/byte ten-byte/word machine.
- off Either FOURB or EIGHTB is on.
- (18) VDM-on Variable-dimension memory. LC1, LC2, and LC3 are not used. Only LCM is expanded.
- NOTE: On the CDC-7600, the maximum memory request is 131 000 words.

- off Use LC1, LC2 and/or LC3 to define LCM space.
- (19) LC1, LC2, and At least one of these flags must be on if VDM is
LC3 off
- LC1 Add 50 000 decimal words to the LCM space.
- LC2 Add 100 000 decimal words to the LCM space.
- LC3 Add 131 000 decimal words to the LCM space.
- Any combination of the LC1, LC2, and LC3 flags may be turned on for maximum of 281 000 decimal words of LCM space.

(2) Model and Processing Options.

Fluid Dynamics

- (1) AEOS-on Use the analytic equation of state.
- off Use the tabular equation of state.
- (2) NOPHASE-on Set the vaporization/condensation rates to zero. The
 PHASE routine is not called.
- off Do vaporization/condensation when appropriate tests are
 satisfied.
- (3) MCPT-on Use the multicomponent phase-transition model.
- off Do not use the multicomponent phase-transition model.
- (4) MATSOL-on Use the SLOR algorithm in the pressure-iteration
 solution.
- off Use the AIM algorithm in the pressure-iteration
 solution
- (5) PLUG-on Use the algorithm for ejection of the rotating plug
 from the reactor head.
- off Do not attempt to model a movable plug.

Neutronics

- (6) URANUS-on Create the point kinetics version of SIMMER-II known as
 "URANUS."
- off Create a space-time kinetics version of SIMMER-II.
- (7) COREXS-on Store all macroscopic cross sections (and finite-
 difference coefficients for the diffusion option) in
 LCM.

- off Store only macroscopic cross sections for the current group in LCM; other groups are stored on disk.
- (8) DIFF-on Create a SIMMER-II version using the time-dependent diffusion theory method.
- off Create a SIMMER-II version incorporating the time-dependent discrete-ordinates transport-theory method.
- (9) FIXUP-on Corrects negative angular and scalar fluxes for the transport theory.
- off No correction of fluxes.
- (10) POSDIF-on A positive-definite differencing operator is used for the transport theory.
- off Negative fluxes are ignored or corrected if FIXUP is on.
- (11) P1APRX-on For the transport option, use an approximate treatment for anisotropic (P1) scattering based on the Beil-Hansen-Sandmier prescription (no effect for the diffusion option).
- off For the transport option, treatment is based on isotropic (P_0) scattering only.
- (12) RHOMAP-on Calculate, print, and store, according to input flags, pointwise reactivity contributions.
- off No separate calculation of pointwise reactivity contributions.
- (13) NEUPREP-on Uses efficiency- and accuracy-enhancing techniques for cross-section calculations. Requires use of the MXS neutronics preprocessor before beginning a transient calculation.
- off Advanced cross-section treatment not used; default method (described in Chap. V) used.
- (14) XSSTAT1 If NEUPREP also is defined, cross-section statistics are calculated (and printed according to input variable IXSPFM), to evaluate vector efficiency.
- (15) XSSTAT2 If NEUPREP also is defined, cross-section timing statistics are calculated to evaluate vector efficiency.

Miscellaneous

- (16) NOTPLT-on Delete all on-line plotting routines. Does not affect postprocessing files.

-off	Include on-line plotting routines.
(17) STAT-on	Include the timing statistics. The user must furnish the subroutine TICHEK.
-off	Do not include the timing statistics.
(18) SPECDMP-on	Save all restart dumps on files TAPEB, TAPEC,..., TAPEZ. Only 25 files allowed per machine period.
-off	Saves only the last two restart dumps on files TAPE21 and TAPE22.

2. Deck Descriptions. The subroutines and functions in the SIMMER-II source file and the functions they perform are listed in Table N-I. The subroutines are listed in the order they appear in the source file, not in the order they are loaded.

The common decks required by the SIMMER-II routines are listed in Table N-II. All variables contained within these common blocks are listed alphabetically and defined in Appendix J. The last word in each common block is a dummy word used to determine the length of the common block for restart reading and writing. Except for the following, the common blocks are brought into core with the main overlay. The common blocks PASS, EOS, and KINET are required only for the fluid-dynamics calculation and are brought into core with the third primary overlay. Common blocks EOS and KINET also are brought in with the restart overlay (4,0) and with common block NITIAL in the fluid-dynamics input overlays (1,6) and (1,8).

Common block SIMCOM is used to store the primary small-core data for both the neutronics and the fluid-dynamics calculations. However, because the data requirements differ for the neutronics and fluid-dynamics calculations, the block lengths differ for the two calculations. Therefore, two common decks are used: IA for the neutronics and KOM for the fluid dynamics. Common block SIMCOM is brought into core with each primary overlay. In overlays (1,0) and (4,0), the length of SIMCOM must be sufficiently large for either the fluid dynamics or the neutronics. The IA common deck is used in the neutronics overlay (2,0), and the KOM common deck is used in the fluid-dynamics overlay (3,0). The data for each SIMCOM common block are stored on disk and transferred to the common block when the corresponding overlays are in core.

Table N-III indicates the control programs for the overlays and the section of the code that they control. PROGRAM OVLIj is the control program for OVERLAY (i,j). The primary/secondary overlay structure is shown in Fig. 24 with the "i"

Fig. 24.
SIMMER-II overlay structure.

Fig. 24 (cont).
SIMMER-II overlay structure.

Fig. 24 (cont).
SIMMER-II overlay structure.

index indicating the primary and the "j" index indicating the secondary overlays. The dashed boxes in this figure indicate the changes in the program structure for the different options discussed in the preceding section. All systems functions that are to be executed before any other instructions, such as initiation of film files, are included in OVLOO. For each operating system, the PROGRAM OVLOO card may have to be rewritten to establish file name usage properly. The rest of the OVLIj programs are dummy programs that only call the first subroutine in the overlay.

3. Program Calculative Flow. As shown in Table N-III the SIMMER-II program consists of five primary overlays: the first reads and processes the input; the second performs the space-dependent neutronics calculation; the third performs the fluid-dynamics calculation; the fourth handles restart files; and the fifth edits execution times. The major calculation effort in SIMMER-II is done in the second and third overlays.

The data used in the neutronics and the fluid-dynamics calculations share the same storage locations in small and large cores. To do this, the data are interchanged between disk and core storage. When neutronics is to be calculated, the neutronics data are brought into both SCM and LCM, and when the fluid dynamics are to be calculated, the fluid-dynamics data are brought into core.

The data that are common to both the fluid-dynamics and the neutronics calculations (that is, material densities, temperatures, and power distributions) are stored in large-core locations before the first location of the noncommon neutronics and fluid-dynamics large-core data.

The overall flow of a complete SIMMER-II calculation is illustrated in Fig. 25. After initializing the system files and setting up the maximum array lengths, the first primary overlay, NPUT1, is called to read and process the input. After initializing constants in NPUT1A, the seventh secondary overlay, NPUT17, is called to initialize further the problem files and to establish whether the problem is a restart or an initial calculation. If the problem is an initial calculation, the space-dependent neutronics input is read, the rest of the neutronics data are initialized, and all the neutronics data are written to disk in preparation for the fluid-dynamics input. As indicated in Fig. 26, the space-dependent neutronics data input routines are skipped completely when the URANUS option is used; instead, the fluid-dynamics data are read immediately. In addition, because there are no space-dependent neutronics data in URANUS problems, it is not necessary to interchange core between the

Fig. 25.
SIMMER-II calculative flow.

Fig. 26.
Overall input calculative flow.

fluid-dynamics and the neutronics calculations. Thus, the fluid-dynamics data always remain in core during URANUS calculations.

The calculative flow for setting up the fluid-dynamics calculation is illustrated in Fig. 27. Three major tasks are performed in the routines SETUP (overlay 1,6), REGINP (overlay 1,8), and DENST. The basic problem specifications are read in subroutine SETUP, and the mesh cell initial conditions are established in subroutine REGINP. In subroutine DENST, the average component densities ρ_m and temperatures T_m are calculated and stored in LCM so that they can be used by the space-dependent neutronics routines for the cross-section calculation. This latter function does not occur with the URANUS option.

If the URANUS option is used, the transient calculation is begun by transferring program control to subroutine TSDRIV as indicated in Fig. 28. Otherwise, the initial flux shapes are obtained by two calls to subroutine GRIND (overlay 2,0) with control then transferred to TSDRIV.

As shown in Fig. 29, the neutronics calculation is controlled by subroutine GRIND for both the initial and the transient fluxes. The first two calls to GRIND are to obtain the initial adjoint and real flux, respectively. The time-dependent real flux is obtained during subsequent calls to GRIND. To control the flow of the calculation in GRIND, the flags ITH, IAD, ITWF, and ITR are used. The flags and the meanings of their values are listed in Table XXII.

For the first call to GRIND, SHLDXS (overlay 2,1) is called to determine the macroscopic cross sections. Calculation flow for the cross-section determination is given in Chap. V. because it is closely related to the method description. If ITWF is input as zero, IAD is set to one and the adjoint flux is to be computed (ITH equals zero for the initial flux calculation). For ITWF provided as one (IAD is set to zero), the real flux is computed. Subroutine OUTER (overlay 2,2) is called for the appropriate flux calculation.

If the adjoint flux was calculated by the first call to GRIND, a second call generates the real flux (IAD is set to zero). No call to SHLDXS is necessary because the cross sections have been determined previously.

After the real flux has been calculated, it is normalized to the initial power and weighting function (adjoint flux or a constant, depending on ITWF) to satisfy the quasi-static constraint condition. These normalizations are performed by a call to subroutine POWCAL, which is called from subroutine PKDRIV (overlay 2,4).

Fig. 28.
Transient calculative flow.

Fig. 28.
Transient calculative flow.

Fig. 29.
Neutronics calculative flow.

Fig. 29 (cont).
Neutronics calculative flow.

TABLE XXII
SUBROUTINE GRIND CALCULATIVE FLOW FLAGS

<u>FORTTRAN Variable</u>	<u>Flag Value</u>	<u>Meaning</u>
ITH	0	Initial flux
	1	Transient flux
IAD	0	Real flux
	1	Adjoint flux
ITR	0	Stationary initial flux
	1	Transient initial flux
ITWF	0	Adjoint flux used as quasi-static weight function
	1	Constant used as quasi-static weight function

The initial amplitude equation parameters are computed in subroutine INPROD, which also is called from PKDRIV.

For third and subsequent calls to subroutine GRIND, ITH is set to one and IAD is set to zero to obtain the transient neutronics solution based on the quasi-static method. Transient cross sections are generated in subroutine SHLDXS and the quasi-static algorithm is used in subroutine PKDRIV. If a flux-shape calculation is required, subroutine OUTER is called for the calculation.

The calculative flow for the flux solution is shown in Fig. 30. Subroutine OUTER controls the fission source iteration, (that is the outer iteration). First the fission source is computed for each iteration. Then subroutine OUTACC is called to normalize and extrapolate the fission source using the Chebyshev acceleration technique. If transient initial conditions are required for the initial real flux solution (ITH = 0, IAD = 0, and ITR = 1), subroutine TRINIT is called from OUTACC to obtain the data adjustment factors. The source to each group is computed next in OUTER, and the group flux is calculated in a call to subroutine INNET or INNET. Convergence of the outer iteration is determined in subroutine OUTACC.

Fig. 30.
Subroutine OUTER calculative flow.

Fig. 30 (cont).
Subroutine OUTER calculative flow.

The solution for each group flux at each outer iteration is calculated in subroutine INNED or INNET. The subroutine used (Figs. 31 and 32) depends on the DIFF processor option. When the DIFF option is on, a neutron diffusion solution is obtained by INNED. For the diffusion solution, the finite-difference coefficients calculated in SHLDXS are placed into the proper matrix locations, and a matrix solution for the fluxes is obtained by FSOL. When the DIFF option is off, a discrete-ordinates neutron-transport solution is calculated by INNET. For the discrete-ordinates solution, convergence of the within-group scattering source iteration is required. For each iteration a space-angle sweep obtains the angular flux, and the rebalance equations are solved for the rebalance factors to be applied to the fluxes.

The calculative flow for the quasi-static method is shown in Fig. 33. The control of the calculative sequence is in subroutine PKDRIV. The flux shape is updated for each reactivity step. Two options are possible, depending on user input controls. Either the flux is extrapolated linearly to the end of the reactivity time step or the flux calculation is performed approximately by calling subroutine OUTER. Next, the amplitude equation parameters are obtained from subroutines INPROD and POWCAL. The amplitude equation is solved in subroutine TSPK, and the spatial distributions of the delayed neutron precursors are updated in subroutine PCALC. If a flux-shape calculation is triggered by the time-step controls, the new flux shape is calculated in subroutine OUTER. The amplitude equation parameters are recalculated with the new flux shape, and the amplitude equation is solved again for the reactivity steps that constitute the flux-shape step. The flux shape then is recalculated until the constraint condition is satisfied.

The control of the fluid-dynamics transient calculation is illustrated in Fig. 34. When the URANUS option is used, the calculation remains in subroutine HYDRO (overlay 3,0) until the problem is finished or the problem execution time is exhausted. When the URANUS option is not used, the calculation cycles between the fluid-dynamics calculation in subroutine HYDRO and the neutronics calculation in subroutine GRIND.

As shown in Fig. 34, the fluid-dynamics calculation is begun by moving the fluid-dynamics data into core. Each time step is begun by choosing the time-step length according to the time-step controls presented in Sec. IV.G. Phase-transition rates are calculated, and their influence on time-step size is determined. If the URANUS option is used, the point kinetics calculation is performed, including an extrapolation of the power through the time step.

Fig. 31.
Neutron diffusion calculative flow in subroutine INED.

Fig. 32.
Neutron transport calculative flow in subroutine INNET.

Fig. 32 (cont).
Neutron transport calculative flow in subroutine INNET.

Fig. 33.
Quasistatic method calculative flow.

Fig. 34.
Fluid-dynamics calculative flow.

Fig. 34 (cont).
Fluid-dynamics calculative flow.

If fluid-dynamics edits and plots are requested, they are made before the fluid-dynamics calculation. The explicit calculation (Sec. IV.D) is followed immediately by the implicit calculation (Sec. IV.E). A new time step then is begun unless the problem final time or the end of the neutronics time step is reached. In the latter case, the mesh cell average densities and temperatures are calculated and stored in LCM for the cross-section calculation.

One of the most important routines for the fluid dynamics calculation is that of the equation of state. The SIMMER-II code is generally run with the AEOS. The AEOS has evolved with many changes and additions since the modified polytropic gas AEOS of SIMMER-I. A schematic flow chart for the current AEOS routine, P4, is given in Fig. 35.

The current P4 routine has five entries. The main entry is to calculate the structure, liquid, and vapor temperatures; the two-phase pressure; and assorted derivatives given the densities and internal energies of the various components. Two tests are made before the calculation proceeds. The first test is on the vapor density. If it is zero, only structure and liquid temperatures are returned. The possibility of a zero vapor density is allowed so SIMMER-II can operate in a mode where only the initial neutronics calculation is wanted. The second test is on the vapor internal energy. If the energy is zero, the code assumes a call has been made from subroutine EOSINP, and the vapor state must be initialized.

If vapor conditions exist, four calculations are performed. The first calculation in the main entry of the P4 subroutine is the vapor temperature iteration required to solve Eq. (IV-35). Three warning messages can be printed. If full convergence is not obtained (generally 10 significant figures), but 7 significant figures are available, a message concerning significant figure difficulties is obtained. Worse, if convergence does not occur, the internal energy initially is raised by 0.1 MJ/kg, a message is printed concerning an AEOS adjustment, and the vapor temperature iteration is redone. If convergence still is not obtained after raising the vapor internal energy, a nonconvergence message is printed, and the maximum temperature from the last two iterations is used. The source of such convergence problems is the triple-valued nature of Eq. (IV-35) when a component's partial pressure is near the critical pressure. The second calculation in the main entry of the P4 subroutine is the setting up of the two-phase pressure, the pressure derivative with respect to density, the vapor saturation temperatures, the heats of vaporization, the vapor heat capacity, and the derivatives of saturation quantities required for the phase

Fig. 35.
AEOS calculative flow.

transition model. The third calculation is that for the liquid temperatures. The iteration for high internal energies is that of Eq. (IV-42). The vapor pressures corresponding to these liquid temperatures then are determined, taking into account superheat considerations. Vapor saturation conditions are initialized for components where no vapor exists. The fourth calculation determines the structure temperatures. Currently, these temperatures are limited by the material's melting temperature.

The DRHODT entry is for the inner iteration of the vaporization/condensation model [Eq. (IV-100)] where the vapor continuity equations must be made consistent with the equation of state. Here $T_{\text{Sat},M}$, $h_{g,M}$, $\partial T_{\text{Sat},M} / \partial \rho_{G,M}$, and $\partial h_{g,M} / \partial T_{\text{Sat},M}$ are calculated to evaluate Eq. (IV-102).

The P4ENER entry calculates the information required to assure convergence of the vaporization/condensation model. In addition to the quantities determined by DRHODT, the quantities $e_{\text{Con},M}$, $e_{\text{Vap},M}$, $\bar{c}_{vG}$, $\bar{c}_{pG}$, and e_G are updated. The P4ENER entry is called once every outer iteration. The logic in the P4ENER entry also is used for initialization at the start of a problem, when the vapor temperature is defined but the vapor internal energy is not.

The P4PRME entry is for calculating the ρ_{Lm} and c_{L1} as functions of the appropriate liquid temperatures. This entry is called from EXFL1 when the new liquid internal energies become available. The liquid temperatures are updated using the same logic as in the main P4 entry.

Finally, the P4AHTI entry is used in mesh initialization. The liquid temperatures are given. For each temperature the calculation is transferred to the appropriate expressions for evaluation of ρ_{Lm} . The fuel sonic velocity, c_{L1} , is obtained from the fuel temperature using Eq. (III-64). The last part of the P4AHTI entry contains the defaults corresponding to the first density test in P4.

In conclusion, except for the liquid density and sonic velocity calculations, the substance of the AEOS routine is contained within the main entry. The ancillary entries are additions to perform that subset of the equation of state required at given points within the fluid-dynamics logical progression.

B. Programming

The programming used in SIMMER-II was designed for the CDC-7600 computer in which high-speed calculations are performed in a SCM. Because SCM is limited, SIMMER-II is divided into overlays that are exchanged between SCM and disk or LCM. The overlay structure, as discussed in Sec. VI.A, is arranged so that a

maximum of three overlay levels are allowed in SCM at any point in the calculation. The main or driver overlay, which is designated (0,0), resides in SCM at all times. Control from the main program dictates whether the neutronics, (2,0), or the fluid-dynamics, (3,0), primary overlay is written into SCM from disk. The primary overlay then calls secondary overlays into SCM as required.

The merging of TWOTRAN-II and KACHINA to form the basis for SIMMER-I and SIMMER-II has resulted in two different programming approaches in SIMMER-II. Because both programs were developed on the CDC-7600, both approaches rely heavily on data transfer between LCM and SCM. However, the treatment of the data transfer differs between the two approaches, primarily because the solution methods are different. Differences also exist in the way SCM data are accessed. An exception to the 7600 programming techniques was made for the cross-section processing using the neutronics preprocessor (NEUPREP option). In this case it was desired to achieve good vector efficiency on computers such as the Cray 1. Because all such machines have large fast memories and because vector efficiency is enhanced by longer vectors, the coding for this option is based on a one-level storage. The programming techniques used are related closely to the methods used and therefore are discussed in Chap. V as well.

1. Fluid-Dynamics Programming. The fluid-dynamics calculation requires access to a minimum of five mesh cells of data: the mesh cell being computed and the four neighboring mesh cells from which material can be convected into the control mesh cell. The operation of SCM with only five resident mesh cells would require a large amount of data transfer between LCM and SCM. However, there is not enough SCM to store the necessary values for all variables for the full mesh. Therefore, three rows (lines) of mesh cells are stored in SCM.

When the LCM option of the preprocessor is in effect, the SIMCOM common block (in addition to the overall mesh and region data) contains the AASC array that provides space for all the data required for the three radial rows of mesh cells. These three rows of data provide a means of reducing SCM space requirements while at the same time reducing data transfer time between SCM and LCM. The calculation in SCM proceeds across a row from the inner to the outer radius of the problem. To facilitate data transfer between LCM and SCM, all variables for a given mesh cell are stored contiguously followed by all the variables of the next mesh cell in the radial direction along the row. These variables are equivalenced to locations in the AASC array and are addressed directly by means of the indexing procedure discussed in the next few paragraphs. The equivalences and the corresponding variables are listed in Table O-I.

Data transfer between LCM and SCM is controlled by subroutine LOOP. In essence, LOOP ensures that three rows of the mesh (j , $j+1$, and $j-1$) are available in SCM for processing and that the corresponding mesh cell pointers (IJ, IJP, and IJM) are set properly to begin the processing of each row. At the end of each row, LOOP must increase one row in j and reset the three pointers accordingly.

The entire mesh is swept beginning at the lower left corner and moving to the right edge, then moving upward row by row. To begin the sweep, the calling program calls the "START" entry in subroutine LOOP. START reads data for the bottom three rows from LCM to the SCM buffer, placing row $j = 1$ in the buffer section designated "row 1/3;" likewise, row $j = 2$ is read into "row 2/3," and row $j = 3$ read into "row 3/3." Rows 1/3, 2/3, and 3/3 are contiguous in SCM, and, like their counterparts in LCM, each contains $NQI = NQ \cdot IP2$ words, where $IP2$ is the number of mesh cells across the row (including the two "fictitious" radial boundary cells) and NQ is the number of words stored per cell.

Having read three rows, the calling program must access the data. This information is provided by setting IJM, IJ, and IJP as pointers to the first words of the $i = 2$ column of mesh cells in each row. Thus, IJM points to an address NQ words into SCM row 1/3, as do IJ in row 2/3 and IJP in row 3/3. With the first three rows of cells read in and the basic pointers set, control is returned to the calling program. The calling program then initiates the double DO loop over the entire mesh.

Secondary indices often are needed to refer to cells located in columns to the left (IMJ) or right (IPJ) of the column identified by IJ. These indices are obtained easily by applying increments or decrements of the variable NQ , the number of variables or the number of storage words per cell to the three primary indices.

Similarly, the calling program can progress across the row from left to right by adding NQ to each index to advance one column at a time. At the end of the row, the indices IJ, IJP, and IJM refer to cells in column $IP1$ that are the outermost calculative mesh cells in the radial direction. A call then is issued to the "LOOP" entry point in subroutine LOOP. The LOOP entry writes row IJM back into LCM because it is not required for the next row of calculations on row $J = 3$. The indices then are reset such that IJ points to the row that was designated 3/3 or the third row of SCM mesh cells data, IJM points to the row designated 2/3 or the second row of SCM mesh cell data and IJP points to the row designated 1/3 or the first row of SCM mesh cell data. The LCM mesh cell data

from row $J = 4$ are read into SCM row 1/3, and control is passed to the calling program for processing the calculation of row $J = 3$. This procedure is repeated until the entire mesh is processed.

An efficiency tradeoff exists between transferring data and using large amounts of SCM storage. The number of variables stored per mesh cell also affects efficiency. With the overlay structure used in SIMMER-II, local variables computed in one overlay are destroyed when a different overlay of the same level is loaded into SCM. Hence, the only means of retaining values of local variables from overlay to overlay is to place these variables in common blocks. If the local variables are cell-dependent, the available options are (1) to add these variables to the cell data block thereby expanding LCM and SCM requirements, (2) to recalculate local variables where required (thereby increasing computational time), or (3) to use certain variables that are not required as data carriers during a given portion of the program. The latter option is the most efficient one, but it requires care in making program changes to assure that necessary data are not overwritten.

A few words concerning the programming in IMFLUD are needed in case the user wishes to modify the solution algorithm. For example, some energy conservation problems could be alleviated by doing liquid component energy convection implicitly with the same velocities as are used in liquid component mass convection. These programming operations are the consequence of SCM storage limits on the CDC-7600.

The first concern is with the eight temporary arrays of size $\bar{I} \cdot \bar{J}$ and the six smaller temporary arrays required for inverting a matrix by the commonly used Approximate Inversion Method discussed in Appendix G, which means extra arrays of potentially 8 600 words given the present dimensional limits on the number of SIMMER-II mesh cells. In the common method of running SIMMER-II (with LCM on the CDC-7600), these arrays (with one exception) are equivalenced to the AASC array. The mesh data are then in LCM while the matrix equations are solved. On machines without LCM, such as the CRAY or IBM system, extra storage is defined for these arrays.

The second concern is with an efficient method of feeding the right sides of the component density vectors into the matrix solution algorithm, MATSOL. A common block of eight $\bar{I} \cdot \bar{J}$ permanent arrays exists for transfer of data between IMFLUD and MATSOL. Seven are needed; there is one extra array for convenience. Potentially 15 vectors of size $\bar{I} \cdot \bar{J}$ exist (8 liquid densities, 6 vapor densities, and 1 vapor energy). These vectors are stored into the eight $\bar{I} \cdot \bar{J}$

arrays and moved to LCM in two or three passes, depending on the number of components. (With evaluation of the work term, the vapor energy requires a separate operation). After the correct matrix coefficients are set up in the permanent arrays, a transfer back to SCM occurs with MATSOL treating each component individually. Solution vectors are transferred from SCM to LCM as they are calculated. Finally, the liquid and vapor solution vectors are transferred as two groups from LCM to SCM for updating the AASC array. In machines without LCM, these operations are performed by memory-to-memory transfers.

2. Tabular Equation-of-State Programming. The tabular equation-of-state routines access tables from a library located on a disk file, magnetic tape, or other mass storage device. At the beginning of a SIMMER-II calculation, subroutine P4TI loads strings of data from the library into the LCM block KLC1. Pointer words to the data are set up in common block S2DIR. The routines P4, E4, GE4, DRHODT, and P4ENER calculate equation-of-state functions and pass them to the fluid-dynamics routines through common-block equations of state.

a. Structure of the SESAME Library. The fundamental unit of information in the SESAME equation-of-state library is a data record. Each record consists of tables that define a specific set of thermodynamic functions. A catalog number is used to identify the type of data in a record. The structure of data records #201, #301, #302, and #401, which are used in SIMMER-II, is given in Table XXIII. A collection of several records preceded by an index record forms a material file. The index gives the identifying material number, the number of data records in the file, and the catalog number of each record.

The library is a collection of material files preceded by a directory. The material files are separated by file marks to simplify searching through the library for a particular material. On an IBM system, these file marks are replaced by one-word ASCII-II records. The directory file gives the number of material files in the library and the material identification for each. The structures of the directory file and the index record are given in Tables XXIV and XXV.

b. LCM Storage and Pointer Words. During the problem setup in SIMMER-II, the routine P4TI loads data from the SESAME library into the AA1 array in LCM block KLC1. The structure of this array is almost identical to that of the library; however, a system of pointer words replaces the record marks and file marks. For each energy component in each field, the pointer words to the

TABLE XXIII
DESCRIPTION OF SESAME DATA RECORDS

Catalog number 201. Basic Data: five real words.

Z - average atomic number

A - average atomic weight

ρ_0 - solid density

B_0 - solid bulk modulus

C_{ex} - exchange coefficient

(The average atomic weight is used in the mixture calculation. Other quantities are not required for SIMMER-II.)

Catalog number 301. Equation-of-state tables: pressure and internal energy as functions of density and temperature NR,NT,(R(I),I = 1,NR), (T(J),J = 1,NT), ((P(I,J),I = 1,NR),J = 1,NT), ((E(I,J),I = 1,NR),J = 1,NT)

NR - number of points on density mesh

NT - number of points on temperature mesh

R - density mesh

T - temperature mesh

P - pressure

E - internal energy

(The 301 data string is not used by SIMMER-II but is reformatted to the 302 form.)

Catalog number 302. Inverted equation-of-state tables: pressure and temperature as functions of density and internal energy NR,NE,(R(I),I = 1,NR), (ET(J),J = 1,NE), (EC(I),I = 1,NR), ((P(I,J),I = 1,NR),J = 1,NE), ((T(IJ),I = 1,NR),J = 1,NE)

NR - number of points on density mesh

NE - number of points on thermal energy mesh

R - density mesh

ET - thermal energy mesh (ET=E-EC)

EC - internal energy at 0 Kelvin

P - pressure

T - temperature (The 302 data string is not kept on the library. Instead, it is generated from the 301 data string at setup, using the routine INV301.)

TABLE XXIII (CONT)

Catalog number 401. Vaporization Curve: NT,(P(I),I = 1,NT), (T(I),I = 1,NT), (RG(I),I = 1,NT), (RL(I),I = 1,NT), (EG(I),I = 1,NT), (EL(I),I = 1,NT)

NT - number of temperatures

P - vapor pressure

T - temperature

RG - vapor density on coexistence line

RL - density of liquid or solid on coexistence line

EG - vapor internal energy on coexistence line

EL - energy of liquid or solid on coexistence line

(When interpolating on a 401 data string, all quantities except EG can be used as independent variables.)

T(NT) - the critical temperature.

TABLE XXIV

DIRECTORY FILE FOR THE SESAME LIBRARY

The directory file consists of two records.

Record 1 consists of three words: N, DATE and VERS

N - number of material files in the library

DATE - date of current version

VERS - version number

Record 2 consists of two arrays - (MATID(I),I = 1,N) and (NWDS(I),I = 1,N)

MATID - table of material identification numbers, in the same order as the material files in the library

NWDS - for each material file MATID(I) and NWDS(I) is the number of words in the index record

TABLE XXV

INDEX RECORD FOR A SESAME MATERIAL FILE

An index record consists of the following word string: MATID, DATE1, DATE2, VERS, NREC, (TBLID(I), I = 1, NREC), (NWDS(I), I = 1, NREC)

MATID - material identification number

DATE1 - date of creation of material

DATE2 - date when material file was last updated

VERS - version number of most recent update

NREC - number of data records

TBLID - table of catalog numbers for the data record, in the same order as they are stored in the material file

NWDS - for each TBLID(I), NWDS(I) is the number of words in the data record

Notes:

1. Routine SMGETI searches through the library for a particular material file and obtains the index. Next, it loads the data records into local storage by calling routine TBFCH.
2. Routine TBFCH uses the index to obtain a particular data record from the library. After reading the desired record, it backspaces to the first word of the material file.

appropriate data records in AA1 are contained in the array LCFW of common block S2DIR. Common block S2DIR has the following structure.

COMMON/S2DIR/LCMX, NRS, LCFW(4,16), IFN.

For the tabular equation-of-state routines, the 16 energy components of SIMMER-II are ordered.

1. Fuel	}	Vapor field
2. Steel		
3. Sodium		
4. Control		
5. Fission gas		

6. Fuel (liquid)	}	Liquid field
7. Steel (liquid)		
8. Sodium		
9. Control		
10. Fuel (solid)	}	Structure field
11. Steel (solid)		
12. Fuel (fabricated)		
13. Fuel (frozen)		
14. Steel (cladding)		
15. Steel (can wall)		
16. Control		

The four words for each component in the array LCFW are as follows.

LCFW(1,I) - SESAME material identification for component I

LCFW(2,I) - first-word location of the 201 data record in array AA1

LCFW(3,I) - first-word location of the 302 data record in array AA1

LCFW(4,I) - first-word location of the 401 data record in array AA1

No tables are loaded from the library for fission gas. In place of the 201 table, the atomic weight specified by the user is loaded into array AA1, and the pointer words for the 302 and 401 tables are set to zero.

The parameter IFN of common block S2DIR specifies the interpolation algorithm in routine RATFN2 (IFN = 0 for the rational function, IFN = 1 for bilinear). LCMX is the last word in the array AA1 that can be used for equation-of-state tables. NRS = 5 is the number of different materials in SIMMER-II (fuel, steel, sodium, B₄C, and fission gas).

In SIMMER-II, the melting temperature and heat of fusion are taken to be independent of pressure. These constants for each material are initialized by DATA statements in P4TI.

c. Calculation of Thermodynamic Functions. Subroutine P4 is the driver routine that computes thermodynamic quantities from the tables in array AA1 and the pointer words in common block S2DIR. A list of the internal variables used in this routine is given in Table XXVI. Routine DRHODT is an abbreviated version of P4 in which certain calculations have been omitted to save computing time. In both routines, density and internal energy are the independent variables used in the equation-of-state computations. The units used in the SESAME library are pressure (GPa), energy (MJ/kg), density (mg/m³), and temperature

TABLE XXVI

INTERNAL VARIABLES USED IN ROUTINE P4

A(I)	Average atomic weight for vapor component I (1 is fuel, 2 is steel, 3 is sodium, 4 is B ₄ C, and 5 is fission gas)
XM(I)	Mole fraction for vapor component I
XW(I)	Weight fraction for vapor component I
RI(I)	Effective density for vapor component I
ECI(J,I), PCI(J,I)	Cold curve for vapor component I
ECI(1,I), PCI(1,I)	Energy and pressure
ECI(2,I), PCI(2,I)	Density derivatives
PI(J,I), TI(J,I)	Pressure and temperature for vapor component I
PI(1,I), TI(1,I)	Pressure and temperature
PI(2,I), TI(2,I)	Density derivatives
PI(3,I), TI(3,I)	Energy derivatives
ABAR	Average atomic weight for vapor mixture
EC, DECDR, PC, DPCDR	Cold curve for mixture
EI	Thermal energy for vapor component I
Y	Array used for temporary storage

(K). Conversion to SIMMER-II units is done in routine SMGETI, which is called from P4TI.

P4 and DRHODT call subroutines LA302B, LA302S, and LA401A. LA302B operates on a data record having format #302, as described above, to compute the 0 Kelvin pressure and energy. These quantities are used in the vapor mixture algorithm. LA302S operates on a 302 data record to define the pressure and temperature as functions of density and internal energy. Routine LA401A operates on a 401 data record to compute variables on the coexistence curve. The user specifies the independent variable with this routine. All three of these basic routines return first derivatives as well as the fundamental quantities themselves.

Routines GE4 and P4ENER also are abbreviated versions of P4. However, the temperature of the vapor mixture, rather than the internal energy, is an independent variable. Because of the complexity of the mixture rule, a Newton's method of iteration is used to invert the temperature-vs-energy relation.

Routine E4 is used to compute initial mesh cell thermodynamic variables during problem setup. For SIMMER-II, there are three branches in E4. A special call to E4 from subroutine REGINP is used to initialize the solidus and liquidus energies (SIESOL and SIELIQ in common-block equation of state) for the five material components. Subsequent calls to E4 are from subroutine EOSINP and depend on the specification of the vapor component densities as indicated by the input parameter IINP. If the vapor-field densities are not specified (IINP = 0), the E4 routine computes saturation properties for specified vapor-field and liquid-field energy components as functions of the specified vapor pressure. If the vapor densities are specified (IINP = 1), the E4 routine computes the vapor pressure and energy for the specified density and temperature.

3. Space-Dependent Neutronics. In the space-dependent neutronics of SIMMER-II, a given variable is available for all mesh cells. This approach is taken because the fission source iteration depends on calculating the neutron leakage for each group for all mesh cells. For efficient data transfer, the data are stored in blocks of IT by JT in LCM and transferred as a block when it is needed in SCM.

a. Small-Core Storage. The main storage arrays for the neutronics calculation are in the SIMCOM common block (common deck IA), which consists of the arrays IA, FIA, and A. All integer variables and pointers are equivalenced to locations in the IA array. These equivalences are shown in Table P-I. The pointer words indicate positions in the container array A used as temporary storage in SCM for the blocks of data required by a neutronics routine. As indicated by the sequencing of the pointer values in subroutine NPUT11, the data storage requirements are determined exactly with arrays contiguous and, in some cases, overlapped. If the user's local computer system will not permit the size of the A array to be the current 21 400 decimal words, he may reduce the A dimension if his problem will fit in a smaller A array.

Pointer words are used to indicate the first word address of each data block. As an example, LFL contains the first word address of the flux block in the A array and A(LFL) is the first word of the flux array.

Variable dimensioning has not been used throughout the neutronics programming in SIMMER-II. In several cases, the pointer word and the A array have been used directly in computations; for example, A(LA5+I-1) has been used instead of A5(I). Also, locations in the A array are sometimes used by different blocks of data as indicated in Table P-I. An example is the Q data block and the SOURCE data block.

Single-word floating-point variables required to execute the problem are equivalenced to locations in the FIA array as shown in Table P-II.

b. Large-Core Storage. The allocation of LCM for the neutronics data is handled in the same manner as small-core storage. Most of the group-dependent arrays are stored in LCM so that the dimensionality is IGM multiplied by the small-core requirement of the array. A complete list of the neutronics LCM storage is given in Table P-III.

4. Cross-Section Storage and Programming. The SIMMER-II neutronics are designed to account for changes in the mesh-cell macroscopic cross sections resulting from changes in mesh-cell compositions and temperatures with time. Variations in material densities and temperatures produce changes in the resonance self-shielding of individual isotopes because of Doppler broadening and background dependencies. These dependencies are modeled by multiplying infinitely dilute isotopic cross sections with shielding factors interpolated from tables representing background cross section and isotopic temperature dependence, thus producing shielded microscopic cross sections. The cell average macroscopic cross sections then are obtained by appropriate mixing of the shielded microscopic cross sections.

In the following it is assumed that the neutronics preprocessor option NEUPREP is not used. The unshielded microscopic cross sections are input from the standard interface file ISOTXS. The input process selects the designed isotopes and sorts the cross sections from isotope ordering to group ordering. Cross sections not requiring shielding are summed at this time if possible. Table XVII shows the symbol, storage pointer, and definition of the different microscopic cross sections created. In addition to the group-dependent data shown in the table, the total thermal yields for fission and capture for each isotope are saved. SIMMER-II has two limitations not present for the general ISOTXS format.

- (1) Spectrum-dependent fission fractions (χ 's) are not permitted.
- (2) Elastic downscatter is limited to one group, that is $\sigma_{e,g \rightarrow g+1}$.

The shielding factors for the microscopic cross sections are input from the standard interface file BRKOXS. As with the input cross sections, shielding factors for the desired isotopes are sorted from an isotope-ordered format to a group-ordered format. Only four of five standard reaction types are retained for the diffusion option (that is DIFF on): (1) capture, (2) fission, (3) transport, and (4) elastic. The three reaction types retained for the transport option (that is, DIFF off) are capture, fission, and elastic. In addition to the shielding factors, the tabulation points for background cross section and isotopic temperature are stored. The full range of tabulation points available and corresponding shielding factors are retained. Because the background interpolation is performed over the natural logarithm of the background cross section, the log of the tabulation points is stored. The temperature tabulation points are converted to Kelvin before being stored. Table XVIII shows the symbol and storage pointer for the different shielding factors and tabulation points.

If the neutronics preprocessor option NEUPREP is used, the above discussion still applies but with some exceptions. First, with this option, materials composed of isotope combinations that remain constant throughout a time-dependent calculation may be used in addition to single isotopes. This capability has an effect on the physics treatment but not on the programming because these materials are treated the same as single "isotopes" for purposes of input and data management and this distinction may be ignored. Second, the shielding-factor interpolation is performed by evaluating prefitted and preselected functions in the case of the background dependence and by using preselected interpolating functions in the case of the temperature dependence. Thus, the data being transmitted in the BRKOXS file are not shielding factors but function coefficients and parameters. The data management is affected only in that the number of coefficients usually differs from the number of shielding factor tabulation points. However, additional information must be transmitted from the neutronics preprocessor to SIMMER-II. Because the preprocessor selects the interpolation functions to be used by SIMMER-II and prepares the coefficients accordingly, flags must be transmitted to SIMMER-II indicating which functions are to be used with which coefficients. Two flag tables are necessary: one for the background interpolation and one for the temperature interpolation. The

BRKOXS file input to SIMMER-II is modified to contain these flag tables and other minor changes required by the new treatment. These changes are described fully in the MXS manual. Because of the changes in the BRKOXS file and the changes in the SIMMER-II data management to facilitate vectorization with the NEUPREP option, two versions of the subroutine to read the BRKOXS file (INFF) exist. The correct version is extracted from the SIMMER-II preprocessor library depending on the definition of the NEUPREP option (on or off).

For both the NEUPREP-off and NEUPREP-on versions, the region-dependent isotopic composition of the material components must be specified in addition to microscopic cross sections and shielding factors. This information is used to determine the isotopic background cross section for the shielding factor interpolation and to mix the shielded microscopic cross sections. Thus, the user inputs mixture atom densities, a mixture region assignment table, and a mixture mass-density table. The mixture assignment table assigns each component in a cross-section region to be a specific mixture. The mass-density table is used to normalize the mixture number densities to obtain isotope densities per unit component density, which when multiplied by the component macroscopic densities, result in the appropriate isotopic macroscopic densities. The mass-density table and the mixture densities should be consistent; that is, the mixture atom densities should be calculated for the material density given in the mass-density table. A material may be assigned to more than one cross-section region as long as the isotopic composition of that material remains constant. Table XXVII indicates the input data required to perform these functions.

For example, suppose the mixture instructions are as follows.

<u>Mixture (Implied)</u>	<u>A(LCM)</u>	<u>A(LMD)</u>
1	0	0.0
1	1	0.047
1	20	0.033
2	0	0.0
2	2	0.75
2	3	0.25
3	0	0.0
3	15	0.004
3	14	0.007

TABLE XXVII

MICROSCOPIC CROSS-SECTION MIXING INSTRUCTIONS

<u>Location of Block or Variable Name</u>	<u>Number of Items</u>	<u>Data Function</u>
A(LDC)	NC*NRXS	Component cross-section identification numbers. These numbers assign a set of mixtures to each component for each cross-section block.
A(LCD)	MT	Theoretical densities (kg/m^3) for each mixture assigned previously.
RXSBD(4*NRXS)	4*NRXS	Cross-section region boundary cells. Enter by region, left boundary cell, right boundary cell, bottom boundary cell, and top boundary cell.
A(LMC)	MS	Numbers specifying the composition of mixtures in terms of input isotopes. [The first name in A(LISONM) is number 1, and so on.] The series of numbers identifying each mixture always is begun with a zero. Null compositions are identified by the zero with no subsequent entries.
A(LMD)	MS	Atom number density (atoms/barnmeter) of the isotopes specified in A(LMC).

For this example, there are $MS = 9$ instructions. In the first three instructions, mixture 1 is composed of 0.047 parts of input isotope 1 and 0.033 parts of input isotope 20. If isotopes 1 and 20 are microscopic cross sections in barns, then 0.033 and 0.047 multiplied by 10^{24} are the component microscopic atom densities. In the second set of instructions, mixture 2 is the result of adding three-fourths of isotope 2 to one-fourth of isotope 3. In the next set of instructions, mixture 3 is composed of a combination of isotopes 14 and 15.

These atom densities are divided by the mixture theoretical densities, A(LCD), before they are used in the following algorithm of subroutine SHLDXS to compute the mesh-cell number densities for each isotope.

```

DO 6 N = 1,NC
NRPT = (NRNO-1)*NC+N
MXS = NA(LDC+NRPT-1)
LIM = NA(LNMN+MXS-1)
IF(LIM.EQ.0) GO TO 6
TEMP1 = A(LCPDT+IK+N-2)
TEMP3 = A(LCPDT+IK+NC+N-2)
LOC = NA(LLMN+MXS-1)
DO 10007 M = 1,LIM
TEMP2 = TEMP1*A(LMD+LOC)
ISO = NA(LMC+LOC)
A(LOC2+ISO-1) = A(LOC2+ISO-1)+A(LMD+LOC)
A(LISDEN+ISO-1) = A(LISDEN+ISO-1)+TEMP2
A(LISTMP+ISO-1) = A(LISTMP+ISO-1)+TEMP2*TEMP3
LOC = LOC+1
10007 CONTINUE
6 LOC2 = LOC2+LNISIP

```

In this example, the variable NRNO has been defined before this section of coding to be the cross-section region assigned to the current mesh cell. MXS indicates the appropriate sequence of mixtures corresponding to the NC components for that mesh cell. The variable LIM is the number of isotopes composing a mixture. TEMP1 contains the component macroscopic density, as determined in earlier coding. LOC indicates the beginning location for isotopic densities, A(LMD), and isotopic identifiers, NA(LMC), for each mixture. The individual isotopic densities are accumulated for all components in the mesh cell. An isotope may be contained in more than one material, or a material may be null (that is, contain no isotopes).

In addition to specifying the mixture atom densities, the user provides a mixture assignment table and a mixture mass density table. The mixture assignment table assigns each component in a cross-section region to a mixture. The mass-density table is used to normalize the mixture number densities to obtain isotope densities per unit component density, which when multiplied by the component macroscopic density, results in the appropriate isotopic macroscopic density. The mass-density table and the mixture densities should be consistent; that is, the mixture atom densities should be calculated for the material density given in the mass-density table. A material may be assigned to

more than one cross-section region as long as the isotopic composition of that material remains constant.

There are two versions of SHLDXS in the SIMMER-II program library, one for NEUPREP on and one for NEUPREP off. The above coding for computation of mesh-cell number densities is virtually the same in both, with differences only resulting from the explicit reference to most arrays in the NEUPREP on version of SHLDXS instead of the use of pointers. However, the remainder of the subroutine is quite different in the two versions because of method differences as well as data management differences. Programming of the different versions is described below.

a. Programming the NEUPREP-Off Version of SHLDXS

A system of pointers controls access to stored microscopic cross sections and their corresponding shielding factors. This system is designed to

1. store data compactly, that is, without "holes" between unequal size blocks;
2. provide inherent flexibility to facilitate changes in the number of data items stored; and
3. permit efficient indexing of calculative loops.

These goals have been achieved but at the cost of reduced readability of the FORTRAN coding. Accordingly, Table XXVIII, which defines the various pointers used, and Fig. 15, which illustrates the calculative flow, help provide a general impression. In addition, the following two illustrations of typical calculative loops are provided to assist users who wish to modify the programming.

In subroutine SHLDXS, the required shielding factors are interpolated to the proper background and temperature values. The data manipulation required in the various loops is as follows.

<u>Code</u>	<u>Description</u>
DO 8 M = 1, LNISIP	Loop over all input isotopes.
.	.
.	.
.	.
.	.
Locate tabular values for the background interpolation and; find KSIGL, the lowest point, and KSIGH, the highest point used.	Corresponding tabular values for the temperature interpolation KTEML and KTEMH have been determined previously (outside the background iteration loop).
.	.

.	.
.	.
.	.
LOCFF = LFFF + NA(LLFFGP + M - 1) + KSIGL - 1	Locates shielding factor, corresponding to KSIGL for isotope M.
.	.
.	.
.	.
.	.
LOCSO = LSIGP + NA(LLSIGT + M - 1) + KSIGL - 1	Locates background tabulation point corresponding to KSIGL for isotope M.
.	.
.	.
.	.
.	.
LOCTP = LSIGT + NA(LLTEMT + M - 1) + KTEML - 1	Locates temperature tabulation point corresponding to KTEML for isotope M.
.	.
.	.
.	.
.	.
DO 16 L = 1, LIFFB	Loop over the required types of shielding factors.
.	.
.	.
.	.
LOCFF = LOCFF + (KTEML - 1) *NBINT	Increment shielding-factor pointer from the first temperature in the block to the desired temperature.
.	.
.	.
.	.
.	.
DO 15 K = 1, (KTEMH - KTEML + 1)	Loop over interpolation temperatures.
.	.
.	.
.	.
.	.
Interpolate shielding factors over background for each temperature.	.
.	.
.	.
15 LOCFF = LOCFF + NBINT	Increment shielding-factor pointer as each temperature block is traversed.
.	.
.	.
.	.
.	.

Interpolate shielding factors over
temperature.

.
.

.
.
.
.

16 LOCFF = LOCFF + (NBTEM - KTEMH)
* NBINT

.
.
.
.
.

Increment shielding-factor pointer to
the end of the block for the current
type.

.
.
.
.
.

8 CONTINUE

After the required shielding factors have been calculated, the shielding factors are applied to the unshielded cross sections. The calculative loop accesses the cross-section data as follows.

L = LC

DO 1 M = 1, LNISIP

Loop over all input isotopes.

.
.
.
.

.
.
.
.

....A(LISTRN + L - 1)...

Reference to the transport cross section.

.
.
.
.

.
.
.
.

...A(LISFIS + L - 1)...

Reference to the fission cross section.

.
.
.
.

.
.
.
.

1 L = L + LISMAX

.

Increments cross-section pointer to the
next isotope.

.
.
.

.
.
.

TABLE XXVIII

SCM STORAGE POINTERS FOR UNSHIELDED MICROSCOPIC
CROSS SECTIONS AND SHIELDING FACTORS

<u>Pointer or Variable</u>	<u>Usage</u>	<u>Definition</u>
LC	$A(LC+L)$, where $L = (I-1)*LISMAX$	Block of microscopic cross sections for one group of isotope I.
LNTEMT	$NA(LNTEMT+I-1)$	The number of shielding factors tabulated at different temperatures for input isotope number I (same as NBTEM).
LLTEMT	$NA(LLTEMT+I-1)$	The total number of shielding factors at different temperatures for all isotopes numbered less than I. (Equal to the summation of previous NBTEMs).
LNSIGT	$NA(LNSIGT+I-1)$	The number of shielding factors tabulated at different backgrounds for input isotope number I (same as NBINT).
LLSIGT	$NA(LLSIGT+I-1)$	The total number of shielding factors at different backgrounds for all isotopes numbered less than I (equal to the summation of previous NBINT's).
LNFFGP	$NA(LNFFGP+I-1)$	The total number of shielding factors stored for input isotope I (equal to $NBINT*NBTEM*LIFFB$).
LLFFGP	$NA(LNFFGP+I-1)$	The total numbers of shielding factors stored for all isotopes numbered less than I (equal to the sum of previous $NBINT*NBTEM*LIFFB$ values).
LSIGT	$A(LSIGT + L-1)$ where $L = NA(LLTEMT+I-1)$	Temperature tabulation points for shielding factors of isotope I.
LSIGP	$A(LSIGP+L-1)$ where $L = NA(LLSIGT+I-1)$	Background tabulation points for shielding factors of isotope I.
LFFF	$A(LFFF+L-1)$, where $L = NA(LLFFGP+I-1)$	All shielding factors for 1 group of isotope I.

These examples, Tables XVII, XVIII, XXVIII and Fig. 15 describe the programming of the cross-section shielding subroutine, SHLDXS. Previous sections in Chap. V have described the physics and calculative sequences involved.

Calculation of each different macroscopic cross section proceeds by cells until the end of a row is reached. The cross sections then are transferred to LCM. After all of the rows have been processed, all cross sections for the group are transferred to disk if the COREXS option is off, otherwise they remain in LCM. The series then repeats until all of the groups have been processed.

The macroscopic cross sections are stored by type and group so that all cross sections of a particular type are stored contiguously. First is a block containing the basic types for a particular group, total, self-scatter, and fission (for the transport option); second are all the downscatter cross sections (if any); and third are component power cross sections. This choice is used for both LCM and disk storage. (See Fig. 15).

The calculation of the macroscopic cross sections was described in Chap. V.B.1. After this calculation in subroutine SHLDXS, the macroscopic cross sections are stored by group on disk or LCM for later retrieval in the neutronics calculation. For group G, the first LMTBLK cross sections are stored in blocks of IT by JT for the flux solution. The blocks of downscatter cross sections are next. The last set of blocks contains the energy deposition cross sections required for the power calculation. These macroscopic cross sections are read into LCM in subroutine CSGEN if COREXS is not defined and then are brought into SCM as needed in the calculation.

b. Programming the NEUPREP-On Version of SHLDXS. This version of SHLDXS need not be explained in the same level of detail as in the previous section because the programming is much more "readable." However, several features are noted in particular. First, the explicit references to pointers have been reduced significantly. To make this new approach compatible with the rest of SIMMER-II, a transition between the "old" approach and the new one is made in the subroutine that calls SHLDXS, that is, GRIND1. Data blocks with variable locations are subdivided, new pointers are created to indicate the various arrays to be used in SHLDXS, and SHLDXS is called with all variable arrays in the argument list. The one-level storage system used is actual rather than apparent. Storage is allocated in both "SCM" and "LCM;" however, no SCM/LCM transfers are performed. Because of the need to manage data in a manner consistent with good vector computing efficiencies, the COREXS option must be defined to use the NEUPREP option.

The philosophy under which SIMMER-II programming was performed in earlier versions has been modified for this option. First, although the programming still is intended to be appropriate to a number of different computers, the previous orientation to CDC-7600 architecture has been eliminated. In fact, the new cross-section programming cannot be used with the CDC-7600 machines. This restriction permitted the change to a one-level storage organization and allowed the use of more storage for temporary variables by taking advantage of the large fast memories available on vector and virtual memory computers. Second, the past restriction on the use of the system-dependent programming has been liberalized. (System-dependent programming previously was used only for unavoidable functions, for example, graphics, and then only by carefully isolating "primitive" functions.) Significant advantage can be gained by using assembly-language routines as vectorization aids or compiler directives to force vectorization or perform conditional vector merges. This has been done in cases where other installations could be expected to have similar options.

For a calculation loop to take advantage of the vector capabilities of the Cray-1 and Cray-XMP computers used at Los Alamos, all storage references must use only constant increments during the loop execution. If this is not the case for a given algorithm, there are two options: the algorithm can be redesigned or GATHER/SCATTER techniques can be used.

GATHER/SCATTER techniques can be explained by the following example of modifying a nonvectorizable loop to obtain a vectorization gain.

```
DO 1 I = 1, N
  J = Index1(I)
  K = Index2(I)
  Result(K) = Function (DATA(J))
1 CONTINUE,
```

where Index1 and Index2 are arrays with indirect addresses of the data array and the result array, respectively. "Function" refers to in-line coding (not a sub-routine) for the desired mathematical operations. Vectorization is accomplished by the following modifications.

```
Call GATHER (N, Buffer1, DATA, Index1)
Do 1 I = 1, N
  Buffer2(I) = Function (Buffer1(I))
1 CONTINUE,
CALL SCATTER (N, Result, Index2, Buffer2),
```

where GATHER and SCATTER are assembly-language routines to optimally perform

data fetching and storing and Buffer1 and Buffer2 are intermediate arrays that have been introduced to gain the constant increment necessary for vectorization of the loop.

Both vectorization techniques have been used where appropriate in the new cross-section coding. We have assumed that all installations will have optimized routines to perform GATHER/SCATTER functions available. As a substitute, these routines are provided in their FORTRAN version.

Conditional vector merge functions also have been used to increase vectorization gains. Because loops containing "IF"-type statements cannot be vectorized except under very limited circumstances, conditional vector merges often provide significant gains. An example of a conditional vector merge is provided by modifying the following nonvectorizable loop.

```
DO 1 I = 1, N
  IF (DATA(I). LE. 0.0) GO TO 1
  Result(I) = ALOG (DATA(I))
1 CONTINUE,
The vectorized version,
DO 1 I = 1, N
  ARG = CVMGN(DATA(I), 1.0, DATA(I))
  Result(I) = ALOG(ARG)
1 CONTINUE
```

is the same as the nonvectorized version except that 0.0 will be inserted into the result array wherever the DATA array is zero. [CVMGN assigns the value 1.0 to the variable ARG if DATA(I) is zero and assigns the value DATA(I) otherwise.]

An additional vectorization is provided by the forced vectorization compiler directive, CDIR\$ IVDEP. This directive is used because the Cray compiler will not vectorize loops that appear to have a "dependency." (An example of a dependency is provided by the loop

```
DO 1 I =1, N
  A(I+1) = A(I)
1 CONTINUE,
```

which violates the requirement of independence of vector elements.) The directive forces the compiler to ignore an apparent dependency and vectorize anyway. (Such directives often are needed because the logic the compiler uses to detect dependencies is deficient.)

A final consideration affecting the new cross-section programming is vector length. As a general rule, the longer the vector, the more efficient the calculation of each vector element. However, there is often a trade-off between longer vectors and excessive temporary variable storage. A timing study was performed to identify the point of diminishing returns for increasing vector lengths.

5. Data Common to Neutronics and Fluid-Dynamics Calculations. Both neutronics and fluid-dynamics data are stored in LCM. To preserve each kind of data while the other calculation is proceeding, data are transferred between disk and LCM using random access disk data transfers. However, some data are common to both the fluid-dynamics and neutronics calculations and must be saved during the data transfer. These common data are stored in LCM locations smaller than the first location of the neutronics and fluid-dynamics large-core data.

The first set of data common to both the fluid dynamics and the neutronics is the set of material average densities and temperatures. These values are calculated in subroutine DENST and stored in JBAR blocks of $2*NC*IBAR$ words in large-core memory with beginning location LECDEN. After the fluid-dynamics data are written to disk and the neutronics data are read into LCM, the densities and temperatures are read into SCM to be used in the cross-section calculation.

The other set of data common to the two calculations contains the internal energy generation rates Q_{Nm} . These are calculated in subroutine POWCAL and stored in JBAR blocks of $NC*IBAR$ words. The beginning LCM location of the internal energy-generation rates is LECENE and is the next word after the density and temperature data. The neutronics and fluid-dynamics large-core data follow the last word of the internal energy-generation data. After the neutronics data are written to disk and the fluid-dynamics data are read into LCM, the internal energy generation rates are read into SCM to be used in a fluid-dynamics calculation.

C. Data Management

Two general kinds of data files are manipulated in SIMMER-II. These are standard files and internal files. Data handling for each of these is discussed below.

1. Standard Files. The standard files consist of two cross-section-related files, four flux-related files, and an S_n constant file (SNCONS). The two cross-section files contain microscopic group cross sections (ISOTXS) and temperature background/selfshielding tables (BRKXXS), respectively. Two of the

flux files are for regular total fluxes (RTFLUX), one for input and one for output. The other two flux files are for the adjoint total flux (ATFLUX), one for input and one for output. These three files are read using the buffering feature of subroutine REED (see below) in which BUFFER IN (N,1) (A,A(L)) is used,

where

N is the logical unit being read,
N = 27 for RTFLUX input,
N = 28 for RTFLUX output,
N = 29 for ATFLUX input,
N = 30 for ATFLUX output,
N = 32 for SNCONS,
N = 34 for ISOTXS,
N = 35 for BRKOXS,
1 indicates binary read,
A is the first word address for the read, and
A(L) is the last word address for the read.

2. Internal Files. The internal files consist of random files, the film file, the input and output files, the dump file, the postprocessor plot files, and the ENCODE/DECODE file.

Random access files are used for transferring data to and from LCM. The random read and write routines are programmed so that the location of the data on disk is found by a sector pointer. The sector pointer is relative to zero for each file. TAPE1, TAPE2, and TAPE11 are the three random files used by SIMMER-II. The files are not used with the URANUS option.

Because the random read and write routines are system-dependent, the user must supply the program to read and write random data. The random read and write routines used at Los Alamos in SIMMER-II are IDONEQ, FORQTS, and IRANRE; equivalent routines must be written for each installation. The routines are described briefly here for completeness.

IDONEQ(IUN)

A function to test completion of random I/O operation on disk file IUN:

IDONEQ = 0 I/O operation incomplete
≠ 0 I/O operation complete.

Subroutine FORQTS (NAME,RQT)

A subroutine to set up a request table RQT and create and open a disk file "NAME" for random disk read and write.

Subroutine IRANRE (RQT,ILCM,LLEN,NSCT,IUN)

Entry Points: IRANRE - Random read into LCM location ILCM, LLEN words beginning at disk location NSCT of disk file IUN
IRANWE - Random write from large-core location ILCM, LLEN words to disk location NSCT of disk file IUN.

Subroutine SHUFFL uses the above routines and transfers data from SCM and LCM to disk and back. Both neutronics and fluid-dynamics data are moved by SHUFFL. The data moved from SCM are in the SIMCOM common block that is shared by the neutronics and fluid-dynamics portions of SIMMER-II. All of LCM except that common to both calculations is transferred to disk between neutronics and fluid-dynamics overlay calls. Density, temperature, and energy arrays are not moved. The flag IPASWR located in the FWBGN1 common block determines the function of SHUFFL at any given time during the calculative sequence.

IPASWR = 1 - Write neutronics data to disk
2 - Read neutronics data from disk
3 - Write fluid-dynamics data to disk
4 - Read fluid-dynamics data from disk

In general, the routines necessary to perform binary reads and writes between disk, LCM, and SCM must be supplied by the user. SHUFFL and many other routines in SIMMER-II use routines REED and RITE in which all system-dependent data transfer calls are isolated. The operation of these routines is described next.
Subroutine REED (N, IREC, VECP, NWDS, MODE)

This standard interface routine performs binary read operations from disk and transfers data from LCM to SCM, where

N is an integer I/O unit number,
IREC is an LCM pointer,
VECP is a SCM array at least NWDS words long,
NWDS is the number of words to be transferred, and

MODE is an integer used as an operation indicator.

The following values of MODE are active in SIMMER-II; all others will cause an error.

- 1 - Transfer real data from LCM to SCM.
- 2 - Binary read.
- 3 - Binary read with the value of NWDS as the first word of the record.
- 4 - Rewind unit N.
- 5 - Binary integer read. IREC is used as an SCM array.
- 6 - Binary integer read with NWDS as the first word of the record. IREC is used as a SCM array.
- 7 - Rewind unit N and space forward NWDS records.
- 8 - Skip IREC(1) words of a record before reading NWDS words from a record.
- 9 - Read binary record as eight-byte words.
- 10 - Read binary record as eight-byte words except the last word, which is a four-byte word.
- 11 - Read first word of the record as an eight-byte word and the remainder as four-byte words.

In addition to these standard meanings for MODE, four additional functions were added for SIMMER-II.

- 12 - Real data, binary-buffered, read to SCM. In this case, IREC is the first word of small core to receive a word, and VECP is the last word of small core to receive a word. The number of words I to be read is

$$I = \text{LOCF}(\text{VECP}) - \text{LOCF}(\text{IREC}) + 1$$

NWDS is an error flag or the number of words read;

NWDS < 0 indicates a parity error

NWDS = 0 indicates end of file read

NWDS > 0 number of words read

- 13 - Binary buffered read to LCM. IREC is the LCM pointer and VECP is not used. NWDS has the same meaning as when MODE = 12.

- 14 - Transfer integer data from LCM to SCM.

- 15 - Integer data binary buffered read to SCM. In this case, IREC is the first word of small core to receive a word and VECP is the last word of small core to receive a word. The number of words I to be read is

$$I = \text{LOCF}(\text{VECP}) - \text{LOCF}(\text{IREC}) + 1$$

NWDS is an error flag or the number of words read;

NWDS < 0 indicates a parity error

NWDS = 0 indicates end of file read

NWDS > 0 number of words read

Subroutine RITE (N, IREC, VECP, NWDS, MODE)

This standard interface routine performs binary write operations to disk and transfers data from SCM to LCM when

N is the integer I/O unit number

IREC is the LCM pointer,

VECP is SCM array at least NWDS words long,

NWDS is the number of words to be transferred, and

MODE is an integer used as an operation indicator.

- 1 Transfer real data from SCM to LCM.
- 2 Binary write.
- 3 Binary write with the value of NWDS as the first word of the record.
- 4 Write three file marks and rewind unit.
- 5 Binary integer write. IREC is used as the SCM array.
- 6 Binary integer write with the value of NWDS as the first word of the record. IREC is used as the SCM array.
- 7 Write a file mark and backspace over the file mark.
- 8 Not used.
- 9 Binary write of eight-byte words.
- 10 Not used.
- 11 Not used.

Again, for SIMMER-II, MODE = 12, 13, 14, and 15 represent functions in addition to the standard usage.

- 12 Real data, binary-buffered, write from SCM. IREC is the first word to be transferred. VECP is the last word to be transferred.
NWDS < 0 parity error
NWDS ≥ 0 number of words transferred
- 13 Binary-buffered write from LCM. IREC is the LCM pointer, and VECP is not used.
NWDS - as input - the number of words transferred.
- as output - < 0 parity error, and
≥ 0 the number of words transferred.
- 14 Transfer integer data from SCM to LCM.
- 15 Integer data binary-buffered write from SCM. IREC is the first word to be transferred. VECP is the last word to be transferred.
NWDS < 0 parity error, and
≥ 0 number of words transferred.

D. Input Description

The input data necessary to use the SIMMER-II computer code are described in Appendix Q in the order they are entered into the code. The data are separated into the following sections.

1. Problem control cards
2. Neutronics input for space-dependent kinetics (skip if URANUS option is selected)
3. Fluid-dynamics input
4. Point-kinetics input used for the URANUS option
5. Restart input

E. Printed Output Description

The output of SIMMER-II can be printed on paper (logical unit 9), on the film file (logical unit 12), or both. Various options in the program determine the quantity and kind of output received. The printed output is separated into the following six sections.

1. Problem control card input edit
2. Space-dependent neutronics input edit
3. Fluid-dynamics input edit
4. Neutronics initial calculation edit

5. Neutronics transient calculation edit

6. Fluid-dynamics transient calculation edit

Each SIMMER-II input variable is printed with a description of the variable and its effect on the problem.

1. Problem Control Card Input Edit. The first line of SIMMER-II output indicates whether the program being run is the point-kinetics version (the URANUS option) or the space- dependent neutronics version. A "URANUS CODE" or "SIMMER CODE" on the first line indicates the appropriate version. In addition, both the date of the current SIMMER-II source and the execution date of the problem are included on this line. The next line indicates the program version again as well as the operating system for which the program version was created. Following this line is a list of the other preprocessor options selected to create the current load module. Values of each problem control variable read from the first four cards of the problem control input then are edited along with a short description.

2. Space-Dependent Neutronics Input Edit. The first section of the neutronics input edit prints the single-word variables that are read by subroutine NPUT11 and that control the space-dependent neutronics. First, the values of miscellaneous integer variables controlling the problem limits, such as number of radial and axial mesh cells and number of neutron energy groups, are printed with a description of variable. These are followed by the edit of the quasi-static method time-step controls. Editing interval data and options are listed next. The initial conditions then are printed, including the initial power and reactivity. Next, the editing of convergence criteria for the outer and inner iterations is concluded by a listing of options for the cross-section shielding calculation.

The second section of the input edit prints the isotope self-shielding factors if IEDXST = 1. The self-shielding factors are printed by isotope; each isotope is listed by group, temperature set, background cross section, and reaction type. No printing occurs for the NEUPREP-on option because only coefficients for the shielding factor functions are input. The MXS preprocessor should be used instead to print shielding factors and other related information. The third section of the input edit prints the isotope cross sections if IEDXST = 2. For each isotope, the cross sections are listed by energy group.

The fourth section of the input edit prints the array variables read by subroutine NPUT15. These arrays include the mesh spacing, the cross-section mixture commands, the delayed neutron data, and the transient initial-condition

data. The final section of the input edit prints the energy group data and the component isotopic compositions (number density per kilogram).

3. Fluid-Dynamics Input Edit. A title line precedes the edit of the fluid-dynamics input. Then the first section describes the problem geometry, including the number of mesh cells and their size. This is followed by a section listing the parameters that control the numerical methods. Convergence precision, donor-cell treatment, and the single-phase/two-phase transition are controlled by these parameters.

This section is followed by variables indicating the problem finish time and the computer time interval for restart dumps. Next, a summary is printed of the fluid-dynamics output options selected for the problem. Included in this section are the time intervals for long prints, postprocessor dumps, and restart dump files. Maximum density change, energy change, and material displacement per postprocessor dump are indicated. This section concludes with an edit of the view factors used in creating the fluid-dynamics three-dimensional plots. The following section describes the various time-step control variables and lists their values. Structure failure parameters are described and edited next. This includes both the criteria to be used when melting structure and the can-wall yield stress vs temperature table that can be used for the test to allow radial motion. Then, if the URANUS option is selected, some point kinetics data are given. The power level, relative power in each material, Doppler coefficient, and number of delayed neutron groups and decay heating groups are indicated.

A table of material properties and analytic equation-of-state data for the five materials (fuel, steel, sodium, control, and fission gas) follows the point-kinetics. The table is organized into solid, liquid, and vapor state properties. In general, these data are used for energy component properties as well as material component properties. The input microscopic densities for the density components follow the material property table.

Exchange-function data come next. Heat-transfer coefficient correlation data are edited first, followed by the liquid-liquid heat transfer multiplier matrix. Then the drag-correlation coefficient is printed, together with the other momentum exchange function parameters. Parameters controlling mass exchange come last, and in the current version of SIMMER-II, they only apply to the vaporization-condensation model.

The parameter region data follow the section on exchange-function data. These data are tabulated by column for each parameter region. Following the parameter region data, the boundary conditions, including the lower boundary velocities, are edited. Included in this edit of boundary conditions is an edit of input data for a post-disassembly expansion if the PLUG option is defined. After the boundary velocities, some additional AEOS information is edited. These edits provide values for the parameters in Sec. III dealing with the liquid internal energy, the gas constant dependent on pressure, the liquid density data as a function of temperature, and the liquid fuel velocity of sound.

The print of the mesh-cell set data comes next in the fluid-dynamics input edit. The identification card for each mesh cell set is written, followed by the mesh-cell set boundaries. The input options for each mesh-cell set are described, and the parameter region for the mesh-cell set is indicated. A table of component macroscopic densities and temperatures is presented next. The table is completed by the velocities and solid particle radii. If separate data for each mesh cell in the mesh-cell set are specified, each different kind of data is edited in a table for the mesh cells.

If the URANUS option is used, the edit of the point-kinetics data concludes the fluid-dynamics input edit. First, the reactivity-vs-time or power-vs-time table is printed. The radial and axial power distributions are edited next. If the point kinetics option is being used, the radial and axial distributions of the Doppler weight function and the reactivity worths for fertile fuel, fissile fuel, steel, sodium, and control material then are tabulated. Edits of the delayed neutron data, point-kinetics initial conditions, and decay heating data conclude this section.

4. Neutronics Initial Calculation Edit. After reading and processing input, SIMMER-II solves for the neutronic initial conditions before beginning the transient calculations. An adjoint solution is obtained first, followed by the real flux solution. In both cases, a summary of the solution progression is printed at each outer iteration. The amplitude equation parameters are printed at the end of the initial calculations, and a print of region averaged isotopic number densities is obtained if INVPRT has been input appropriately.

The outer iteration summaries are the same for both the adjoint and real solutions if a stationary (eigenvalue) solution to the initial state is sought. If a transient (inhomogeneous) solution is sought, a summary of the neutronics data adjustment factors is included in the summary. Finally, if the transport

theory option is used (DIFF off), a summary of the inner iteration history is included for each outer iteration.

The following format is used for the outer iteration summaries.

Line 1

The current k_{eff} estimate and the ratio of the current iteration k_{eff} to the previous iteration k_{eff} are printed if a stationary-state solution is sought. The ratio of successive integrated fission sources (an analogous quantity) are printed if a transient-state solution is sought.

Line 2

The total fission source error (sum of the differences between mesh cell fission sources for successive iterations) and the mesh cell fission source error (maximum deviation from unity of the mesh cell ratio of fission sources for successive iterations).

Line 3

If the current iteration is accelerated, the order of Chebyshev polynomials and the dominance ratio used in the current series of accelerated iterations are printed. If the iteration is unaccelerated, the dominance ratio estimate is printed.

Line 4

If the current iteration is accelerated, the current estimate of the dominance ratio and the rate of convergence for the current series of accelerated iterations are printed.

Line 5 - Line 7

If the transient neutronics initialization option is used (ITR = 1) and the stationary state real solution is being sought, the following data adjustment factors are printed.

- (a) Velocity factor - the ratio of the current group velocities to the input velocities
- (b) Precursor source factors - for each delayed neutron group, the ratio of the input delayed neutron precursor concentration to the calculated stationary state value
- (c) Physical beta factors - for each delayed neutron group, the ratio of the calculated delayed neutron fraction to the input delayed neutron fraction

Line 8

If the transport option is used (DIFF off), the following inner iteration summary is printed for each group.

- (a) The total number of rebalance iterations
- (b) The convergence precision required for the rebalance iterations
- (c) The total number of scattering-source (inner) iterations

(d) The estimated error achieved by the inner iteration

The following format is used for the isotopic number density print. For each mesh cell and isotope, the isotope name, the atom density for each component, the total atom density for all components, and the total number of atoms are printed.

If the NEUPREP cross-section option is defined, information needed to evaluate the vector efficiency of the SHLDXS routine can be obtained by defining XSSTAT1 and XSSTAT2 in the preprocessor step and setting the cross-section statistics output flag (IXSPFM) in the SIMMER-II input. For XSSTAT2, a timing routine must be provided by the user that approximates the CPU usage (system overhead and I/O eliminated.)

If XSSTAT1 is defined, the first section printed shows the distribution of vector lengths involved in the background and temperature interpolation calculations. A calculative vector is formed by choosing cells in which a given isotope is present within a particular temperature range. Several such vectors may be needed in the background interpolation step, depending on the type of temperature interpolation to be performed in a later step; thus, this information cannot be used to determine an accurate overall average vector length. However, a good impression of the general vector efficiency can be obtained.

Further printing is present only if XSSTAT2 is defined. The second section shows statistics for the background interpolations. Both the relative frequency and relative CPU usage of each type of background interpolation is given. The different types are numbered in the same way as the priority described in Chap. V. The third section gives the statistics for the temperature interpolations. The typed numbers correspond to the formulas given in Chap. V. Finally, the fractional time for mixing macroscopic cross sections is given.

5. Neutronics Transient Calculation Edit. For the neutronics transient calculation, the time-step controls are monitored for both the reactivity update step and the flux-shape steps. If a shape calculation is performed, a summary of the shape calculation solution progression is printed. In addition, a print of the isotopic atom densities is given at each reactivity update step if INVPRT has been input appropriately.

The shape step is estimated by checking 10 different criteria. The smallest step is selected, and the value and criteria used are printed. The message printed depends on which of the following criteria produced the smallest step.

(1) Previous shape step

- (2) Change in leakage
- (3) Number of reactivity steps
- (4) Total problem time
- (5) Maximum shape step
- (6) Change in flux shape
- (7) Change in reactivity
- (8) Minimum shape step

In addition, the shape-step values for the first seven of these criteria always are computed and listed, permitting comparison of the effects of different criteria.

The reactivity step is monitored in an analogous fashion. In this case, the following six criteria are checked, with a printing of the values used and the values of those not used.

- (1) Previous step
- (2) Change in reactivity
- (3) Shape step
- (4) Maximum reactivity step
- (5) Change in amplitude
- (6) Change in inverse period

The outer iteration summary for the transient flux-shape calculation is similar to that used for the initial flux calculation. Because an inhomogeneous calculation is performed, only the fission source error estimates and the Chebyshev acceleration parameters are printed. For the transport option (DIFF off), the inner iteration summary also is printed.

At reactivity steps, an outer iteration summary (and an inner iteration summary if DIFF is off) is printed if the flux shape update option (IFXUDL > 0) is being used. Amplitude equation parameters necessary to assess the progression of the transient are printed as follows. The reactivity and amplitude calculated from the updated amplitude equation solution (based on the current cross sections and flux shape) are printed along with other amplitude equation parameters such as the prompt neutron generation time, inverse period, and effective delayed neutron fractions. Results of the time-step estimation are printed next, as described above. The reactivity projected to the next reac-

tivity step and the corresponding amplitude from an amplitude equation solution follow. Finally, a summary of the energy conservation imbalance between the neutronics and fluid dynamics solutions is printed. This imbalance is attributable to the reactivity changes at the end of reactivity and shape steps (no iteration between fluid dynamics and neutronics is performed) and thus is divided into a reactivity step component and a shape-step component.

At flux-shape steps, a final reactivity step is taken if the flux-shape update option is used, but only the recalculated reactivity (based on the reactivity step update) and amplitude are printed. For the gamma iteration required for the shape step, an outer iteration summary leading to a converged shape solution is printed. If the gamma iteration is not converged (to unity), the quasi-static constraint integral, the reactivity, and the amplitude are printed, and the next flux-shape calculation (with printed summaries) is performed. After each gamma iteration, the series of reactivity steps composing the current flux-shape step is recalculated based on an updated value for amplitude equation parameters. However, the reactivity steps are printed only for the final gamma iteration. All amplitude equation parameters are printed at each of the recalculated reactivity steps. Also, after each gamma iteration, the delayed neutron source of the shape equation is integrated and compared with that of the amplitude equation. The results are printed, and the delayed neutron precursor concentrations of the shape equation are scaled to agree with the amplitude equation quantities. The error in the gamma iteration (constraint constant) is printed next along with the amplitude-equation delayed neutron precursor densities. Finally, the printing described previously for reactivity steps is performed before taking the next reactivity step.

6. Fluid-Dynamics Transient Calculation Edit. The fluid-dynamics transient calculation has four different kinds of output.

- (1) Time-step edits
- (2) Long prints controlled by the input parameter LPR
- (3) Short print summations controlled by the input parameter ISPR
- (4) Diagnostic information and guidance on spurious results

a. Fluid-Dynamics Time-Step Edits. For each time step (cycle), a single line is printed to summarize the results. That line indicates

- (1) time (T),

- (2) time-step number (CYCLE),
- (3) time-step size (DT),
- (4) number of single-phase mesh cells (NUMINC),
- (5) number of pressure iterations (ITP),
- (6) number of nonconverged cells (CELLS),
- (7) number of vapor component iterations (ITRO),
- (8) time-step control number (STEP), and
- (9) CPU time for the current time step (CP).

Items (6) and (7) are currently obsolete. No nonconverged cells can be allowed to exist. If the pressure iteration does not converge, the problem will be terminated. Also, the component densities now are obtained from subroutine MATSOL rather than using the SIMMER-I cell-by-cell iteration on the vapor components. If the URANUS option is used and IPOWER = 0 (point kinetics), the next two lines summarize the current power and reactivity contributions at this time. The first line contains

- (1) total relative power (PHI),
- (2) prompt fission relative power (POWFS),
- (3) decay heat relative power (POWDK),
- (4) time integral of the relative power (PINTG),
- (5) driving reactivity (REAPGM), and
- (6) total reactivity (REATOT).

The second line contains

- (1) Doppler feedback reactivity (READOP),
- (2) fertile fuel displacement reactivity (REAFER),
- (3) fissile fuel displacement reactivity (REAFIS),
- (4) steel displacement reactivity (REASTR),
- (5) sodium displacement reactivity (REACOL), and
- (6) control displacement reactivity (REACTL).

All the above reactivities are edited in dollars. If IPOWER > 0 for the URANUS option, only the total relative power and the power integral are edited.

If the time-step-control edit flag, IEDTSC, is not zero, then the values are printed for each of the fluid-dynamics time-step controls for the current time step. For those time-step controls that are dependent on the dynamics occurring within each mesh cell, the mesh-cell number that controlled the time step also is indicated. The three lines of time-step controls are as follows.

Line 1

- (1) Initial time step, DTL(1);
- (2) long print edit, DTL(2);
- (3) Courant conditions, DTL(3), and the mesh cell initiating the Courant time-step control; and
- (4) vapor volume fraction, DTL(4), and the mesh cell initiating the vapor volume-fraction time-step control.

Line 2

- (5) Maximum time-step increase, DTL(5);
- (6) maximum time-step size, DTL(6);
- (7) pressure iterations, DTL(7), and the mesh cell initiating the pressure-iteration time-step control; and
- (8) phase transitions, DTL(8), and the mesh cell initiating the phase-transition time-step control.

Line 3

- (9) High reactivity limitation (space-time kinetics), maximum power change (point kinetics), DTL(9);
- (10) maximum time before reactivity or shape update (space-time kinetics), maximum reactivity change (point kinetics), DTL(10);
- (11) maximum vapor energy change, DTL(11), and mesh cell initiating the time-step control for the maximum vapor energy change; and
- (12) maximum liquid energy change, DTL(12), and the mesh cell initiating the time-step control for the maximum liquid energy change.

b. Fluid-Dynamics Long Prints. The fluid-dynamics long print provides a table of values of the many mesh-cell-dependent variables for each mesh cell. The printed table is controlled by the input parameters LPR, the devices to be used for the output; LINE(N), what line is to be printed; and NODE and NODES(N), the cells to be printed. See cards 4 and 5 of the fluid-dynamics input for a description of these variables.

Line 1

- (1) External radial mesh index, (I);

- (2) external axial mesh index, (J);
- (3) vapor-field volume fraction, (ALPHG);
- (4) vapor-field radial velocity component, $U_{G,i+1/2}$ (UG);
- (5) vapor-field axial velocity component, $V_{G,j+1/2}$ (VG);
- (6) fertile-fuel vapor macroscopic density, (RGBR1);
- (7) fissile-fuel vapor macroscopic density, (RGBR2);
- (8) steel vapor macroscopic density, (RGBR3);
- (9) sodium vapor macroscopic density, (RGBR4);
- (10) control vapor macroscopic density, (RGBR5), and
- (11) fission gas macroscopic density, (RGBR6).

Line 2

- (1) Liquid-field volume fraction, (ALPHL);
- (2) liquid-field radial velocity component, $U_{L,J+1/2}$ (UL);
- (3) liquid-field axial velocity component, $V_{L,j+1/2}$ (VL);
- (4) liquid fuel temperature, (TL1);
- (5) liquid steel temperature, (TL2);
- (6) liquid sodium temperature, (TL3);
- (7) liquid control temperature, (TL4);
- (8) solid fuel particle temperature, (TL5); and
- (9) solid steel particle temperature, (TL6).

Line 3

- (1) Structure-field volume fraction, (ALPHS);
- (2) liquid fertile fuel macroscopic density, (RLBR1);
- (3) liquid fissile fuel macroscopic density, (RLBR2);
- (4) liquid steel macroscopic density, (RLBR3);
- (5) liquid sodium macroscopic density, (RLBR4);
- (6) liquid control macroscopic density, (RLBR5);
- (7) macroscopic density of fertile fuel in solid fuel particles, (RLBR6);

(8) macroscopic density of fissile fuel in solid fuel particles, (RLBR7);
and

(9) macroscopic density of steel in solid steel particles, (RLBR8).

Line 4

- (1) Fabricated fertile fuel macroscopic density, (RSBR1);
- (2) fabricated fissile fuel macroscopic density, (RSBR2);
- (3) frozen fertile fuel macroscopic density, (RSBR3);
- (4) frozen fissile fuel macroscopic density, (RSBR4);
- (5) cladding macroscopic density, (RSBR5);
- (6) subassembly can-wall macroscopic density, (RSBR6);
- (7) solid control macroscopic density, (RSBR7);
- (8) intragranular fission gas macroscopic density (RSBR8); and
- (9) intergranular fission gas macroscopic density, (RSBR9).

Line 5

- (1) Pressure, (P);
- (2) solid fuel particle radius, (RPSF);
- (3) solid steel particle radius, (RPSS);
- (4) vapor mixture temperature, (TG);
- (5) fabricated fuel temperature, (TS1);
- (6) frozen fuel temperature, (TS2);
- (7) cladding temperature, (TS3);
- (8) subassembly can-wall temperature, (TS4); and
- (9) solid control temperature, (TS5).

Line 6

- (1) Mass-transfer rate of fabricated fuel to liquid fuel, (FS1);
- (2) mass-transfer rate of frozen fuel to liquid fuel, (FS2);
- (3) mass-transfer rate of cladding to liquid steel, (FS3);
- (4) mass-transfer rate of subassembly can-wall to liquid steel, (FS4);
- (5) mass-transfer rate from solid control to liquid control, (FS5);

- (6) mass-transfer rate from intragranular fission gas to intergranular fission gas, (FS6);
- (7) mass-transfer rate from intergranular fission gas to the vapor field, (FS7);
- (8) mass-transfer rate from solid fuel particles to liquid fuel, (FL1); and
- (9) mass-transfer rate from solid steel particles to liquid steel, (FL2).

Negative values for FS2, FS3, FS4, FS5, FL1, or FL2 imply freezing or mass transfers from the liquid components to the structure and particle components.

Line 7

- (1) Mass-transfer rate from fuel vapor to liquid fuel, (GAMCE1);
- (2) mass-transfer rate from steel vapor to liquid steel, (GAMCE2);
- (3) mass-transfer rate from sodium vapor to liquid sodium (GAMCE3);
- (4) microscopic liquid fuel density, (RHOLFU);
- (5) energy-transfer rate to fabricated fuel as a result of heat and mass transfer, (QS1);
- (6) energy-transfer rate to frozen fuel as a result of heat and mass transfer, (QS2);
- (7) energy-transfer rate to cladding as a result of heat and mass transfer, (QS3);
- (8) energy-transfer rate to subassembly can-wall as a result of heat and mass transfer, (QS4); and
- (9) microscopic liquid steel density, (RHOLST).

Vaporization is represented in SIMMER-II as a negative mass-transfer rate. The microscopic steel density is printed instead of the solid control energy transfer.

Line 8

- (1) Energy-transfer rate to liquid fuel because of heat and mass transfer, (QL1);
- (2) energy-transfer rate to liquid steel because of heat and mass transfer, (QL2),
- (3) energy-transfer rate to liquid sodium because of heat and mass transfer, (QL3);
- (4) energy-transfer rate to liquid control because of heat and mass transfer, (QL4);

- (5) energy-transfer rate to solid fuel particles because of heat and mass transfer, (QL5);
- (6) energy-transfer rate to solid steel particles because of heat and mass transfer, (QL6);
- (7) energy-transfer rate to the vapor mixture because of heat and mass transfer, (QG);
- (8) liquid-structure momentum-exchange function, (FIJL); and
- (9) vapor-structure momentum-exchange function, (FIJG).

Line 9

- (1) Liquid-vapor momentum-exchange function, (KIJ);
- (2) hydraulic diameter for the cell, (DH);
- (3) liquid-vapor velocity difference for droplet radius calculation, (VLMGN);
- (4) normalized internal energy generation rate in fertile fuel (ENI);
- (5) normalized internal energy generation rate in fissile fuel, (EN2);
- (6) normalized internal energy generation rate in steel, (EN3);
- (7) velocity of sound in liquid fuel, (VELSOD);
- (8) microscopic density of liquid sodium, (RHOLNA); and
- (9) can-wall failure indicator, (FAIL3).

Line 10

- (1) Liquid fuel droplet radius, (RPLF);
- (2) liquid steel droplet radius, (RPLS);
- (3) liquid sodium droplet radius, (RPLNA);
- (4) liquid control droplet radius, (RPLC);
- (5) fuel saturation temperature following the vaporization-condensation calculation, (TSATF);
- (6) steel saturation temperature following the vaporization-condensation calculation, (TSATS);
- (7) sodium saturation temperature following the vaporization-condensation calculation, (TSATNA);
- (8) control saturation temperature following the vaporization-condensation calculation, (TSATC); and

(9) vapor temperature following the vaporization-condensation calculation, (TGN).

c. Short Print Summations. Summations of the mass, momentum, and energy are presented after every long print if the input variable ISPR is one. If ISPR is two, a summation is provided for each time step. The summation is grouped into five categories.

(1) Volume-fraction summations

(2) Mass summations

(3) Momentum summations

(4) Energy summations

(5) System balance

The volume-fraction summary lists the average volume fraction for each field, the average nonflow volume fraction, and the average total volume fraction, which always should be one.

Five summary columns are printed for the mass and energy summations. The first three summarize the density component mass and energy integrals over all mesh cells for the vapor, liquid, and structure fields. The next two columns contain the net flows out the mesh boundaries for each of the density components in the vapor field and the liquid field. A total for each field is accumulated at the bottom of each column. The momentum summary consists only of the radial, axial, and total summations for each of the three fields; no outflow summary is included.

The system balance compares the total energy contained in the computing mesh with the total initial energy, the energy lost through the boundaries, and the energy added through nuclear heating. A similar comparison is made for the total mass in the computing mesh. Finally, information is provided for use in post-disassembly problems. This information is the cover gas volume, the impulse to the head, the peak pressure on the head with its time and location, and the value and time of the peak average pressure on the head.

d. Diagnostic Information and Guidance on Spurious Results. Several messages can be printed for diagnostic information during the course of a SIMMER-II run. These messages include both the initialization and transient portions of the code. The fluid-dynamics messages in the code initialization section are as follows.

- (1) A warning is printed on paper output if the mesh boundaries indicated in any mesh point set exceed the maximum dimensions (IBAR and JBAR) defined for the problem.
- (2) If the vapor volume fraction is less than or equal to zero because of excessive liquid, a warning is printed on paper output, and the liquid component densities are reduced so that the cell will behave as an ordinary single-phase cell.
- (3) If the vapor volume fraction is less than or equal to zero because the structure volume fraction is greater than one, a warning again is printed on the paper output. An attempt is made to reduce the structure density; however, current SIMMER-II programming does not perform this reduction correctly.
- (4) Additional warnings appear on paper output if either the total liquid or vapor densities are less than zero. Although resets on the sodium liquid and vapor densities to 10^{-10} kg/m³ are performed in such cases, such problems usually indicate other input inconsistencies that will cause code failure.
- (5) If the input variable ICOMPO does not correspond to energy components 1 through 4, or corresponds to a zero liquid density component, a warning message is printed and ICOMPO is adjusted to correspond to a liquid that is present. The procedure is followed only if the input variable IINP is less than one.
- (6) A warning message is printed on paper output for nonconvergence of the iteration to define any vapor-state gas constant.
- (7) A diagnostic message is printed if the liquid component temperatures are above their saturation values and the input variable ISAT is greater than zero. This message informs the user that the liquid temperatures have been reset to saturation values.

The fluid-dynamics messages in the transient section of the code are as follows.

- (1) Three possible messages can come from subroutine HYDRO indicating early problem termination. Only the termination because of an excessively small time step is currently relevant. This message is preceded by 10 "*" characters and appears only on paper output. Generally, this message means that divergent results are being calculated such that the problem cannot be continued. Two long prints precede problem termination.

- (2) If PLUG is defined, diagnostic information on the problem results is edited on each time step. The variables printed include three types of data: plug data, material ejection data, and structural loading data. The plug data include JPLUGI, the node location of the lower plug interface; FPLUG, the force on the plug (N); APLUG, the acceleration of the plug (m/s^2); VPLUG, the velocity of the plug (m/s); and ZPLUG, the location of the plug relative to the initial position (m). The data describing material ejection from the reactor vessel into the containment include TOT SOD, total sodium ejected (kg); LIQ FL, liquid fuel ejected (kg); PAR FL, particulate fuel ejected (kg); VAP FL, fuel vapor ejected (kg); and TOT SS, total steel ejected (kg). The data on structural loads are F HEAD, up-force at the head (N); F CORE, down-force at the core (N); F SHLD, down-force at the shield (N); P WALL, pressure at the upper vessel wall (Pa); and P HEAD, pressure at the center of the head (Pa).
- (3) Various significant changes in reactor conditions can lead to a neutronics update if URANUS is not defined. These conditions are excessive changes in reactor power, fuel internal energy, steel internal energy, sodium internal energy, fuel density, steel density, and sodium density. A long print also triggers a neutronic update to provide consistency between the hydrodynamics and neutronics if $\text{LPR} > 0$. Satisfaction of any of these conditions is indicated by a printed message. An additional message is printed if these updates are overridden by setting the input variable IQUASI greater than one.
- (4) A warning message is printed (paper only) if the iteration to redefine the gas constant does not converge when a cell is changed from a single-phase state into a two-phase state. Twelve "*" characters precede this message.
- (5) When can-wall failure by the pressure-dependent failure criterion occurs, a message is printed. The cell location is identified, the pressures leading to failure are printed, the relevant yield stress and can-wall temperature are identified, and the fractional thickness of the failing can wall is shown.
- (6) A separate message is printed to identify can-wall failure (radial motion) induced by can-wall melting. Here only the cell location is identified along with the pressures of the two adjacent radial cells.

- (7) Whenever the iterative matrix solution for the pressure iteration is directed to test for a change in the attempted mode of convergence (row or column), the number of iterations and the error in the converged solution are printed for both the row and column iterations.
- (8) If the iterative matrix solution for the pressure iteration cannot converge, a message is printed and the problem is terminated. The usual difficulty here is too large a time-step size, resulting in a matrix that is too ill-conditioned for the current methods to invert.
- (9) If the overall pressure iteration does not converge, a message is printed indicating the number of cells that did not satisfy the convergence criteria. Also, the problem is terminated with a full print following recalculation of the exchange functions.
- (10) If the vapor volume fraction from a nominal two-phase cell is calculated to drop below $-\alpha_0$ following the pressure iteration, a message is printed indicating the cell in question and the new vapor volume fraction. Then the flag is set to terminate the problem with a full print following recalculation of the exchange functions. In this case, the problem generally may be restarted. However, care should be exercised in interpreting the results because of the high pressures that will be obtained on the next cycle.
- (11) If the vaporization-condensation model does not converge, a message is printed indicating that a problem exists with the saturation temperature. The cell location of the problem is given, and three additional variables are printed. The variable NITNO is the overall iteration index; a value of 100 generally indicates problems with the vapor energy equation. The variable IRHOIT is the iteration index for the inner iteration on the conservation equations and the equation of state; a value of 100 generally indicates some problem with the equation of state. Either convergence difficulties are being experienced because of the production of a near critical pressure condition for some component, or the only vaporizing component could be one with extremely low vapor pressure where the saturation density is extremely nonlinear as a function of temperature. The variable DETERM is the quantity under the square root sign in the solution of the vapor energy equation. If DETERM is negative, the vapor energy equation does not possess a solution expressible as a real temperature.

- (12) The analytic equation of state possesses a hierarchy of messages indicating increasing difficulty in obtaining a solution for the vapor-field temperature. If full convergence cannot be obtained, the usage of excessive time is printed. If convergence does not occur, generally the triple-valued character of the function being solved is the problem. The vapor energy is raised by 10^5 J/kg, a message is printed indicating an equation-of-state adjustment, and the iteration is restarted. If convergence still does not occur, a message indicating nonconvergence is printed, and the highest temperature from the last two iterations (which we hope to be positive) is used for subsequent calculations. Results may be of questionable validity when this occurs.
- (13) A diagnostic message is printed if the liquid temperature iteration does not converge in the equation of state.
- (14) A diagnostic message is printed if the iteration for the particle radius based on the Weber number does not converge. Generally, this indicates some conceptual difficulty with the formulation of the problem being attempted.
- (15) A message is output from the tabular equation of state if a requested material is not found in the library.

F. Plotted Output

The SIMMER-II output can include information on 35-mm or 105-mm microfiche film. It is also possible to create 16-mm motion pictures directly from SIMMER-II. However, this latter feature is not recommended because each movie of a dependent variable requires redoing the calculation, resulting in wasted computer time. The postprocessor feature should be used instead because it provides the capability to make movies without rerunning the SIMMER-II program itself. In addition to plots, the printed output discussed in the previous section can be obtained on 35-mm or 105-mm microfiche film.

The kind and quantity of fluid-dynamics plots obtained on film is controlled by the input variable LPR. The use of this variable is indicated under Card No. 2 of the program control input. The frequency of fluid-dynamics plots is controlled by the long print time intervals, DTO and DTOC. Three types of plots are available. For two-dimensional calculations, both contour and three-dimensional perspective plots can be used to display the fluid-dynamics

results. Two-dimensional plots are obtained for one-dimensional fluid-dynamics calculations.

The fluid-dynamics plots consist of velocity vector plots for the vapor and liquid fields and contour plots, perspective plots, or both for selected mesh-cell variables.

Velocity vector plots are provided separately for both the vapor and liquid fields. Velocity component vectors are plotted originating from the midpoint of their interface with a line length proportional to their magnitude. The presence of intact can-wall on a radial mesh cell interface by definition has a radial velocity component of zero. To point this out in the velocity vector plot, a solid line is plotted on the radial interface having an intact can-wall. The solid line disappears following can-wall failure, and subsequent motion across a radial interface is indicated by a velocity vector having a radial component.

The 52 different mesh cell variables listed in Table XXIX are plotted on contour plots, prespective plots, or both. If the difference between the maximum and minimum values (excluding zero) of a variable over the mesh is less than 10^{-10} , the variable is not plotted. Before a variable is plotted, its values are stored in the temporary array, CQ, and equivalenced to the AASC block. For the contour plots, linear interpolation between mesh cells is used to determine points of equal value. These points then are connected with vector segments to obtain the contour lines. The contour plots have a constant increment.

The scaling of the contour plots was taken from the KACHINA program. All film output devices are treated as an SC 4020 having a matrix of 1024 by 1024 raster points on the cathode ray tube (CRT) face. For the 4020, the origin of the x-y coordinate system is the upper left corner of the frame, with the values of the two raster point indices increasing down and to the right. This coordinate system differs from the fluid-dynamics coordinate system, where the lower left corner is the origin. A conversion is required to translate a point on the physical mesh to the corresponding 4020 frame position.

Before performing this conversion, the top 16 raster points and the bottom 108 raster points are reserved for problem identification, problem time, cycle number, and other information. In addition, the frame width is reduced by two raster points on the right side to ensure frame separation. Thus, the plotting area is 900 by 1022 raster points. Within this rectangular region, the size of the plot is maximized, maintaining a true proportion of the height to width of

TABLE XXIX
PLOTTED FLUID-DYNAMICS MESH-CELL VARIABLES

<u>Plot Number</u>	<u>Film Identifier</u>	<u>Mathematical Symbol</u>	<u>Description</u>
1	ALPHG	α_G	Vapor-field volume fraction
2	ALPHS	α_S	Structure-field volume fraction
3	ALPHL	α_L	Liquid-field volume fraction
4	P	p	Pressure
5	RSFBR1	$\bar{\rho}_{s1}$	Fabricated fuel macroscopic density
6	RFSBR2	$\bar{\rho}_{s2}$	Frozen fuel macroscopic density
7	RSBR1	$\bar{\rho}_{s1}$	Fabricated fertile fuel macroscopic density
8	RSBR2	$\bar{\rho}_{s2}$	Fabricated fissile fuel macroscopic density
9	RSBR3	$\bar{\rho}_{s3}$	Frozen fertile fuel macroscopic density
10	RSBR4	$\bar{\rho}_{s4}$	Frozen fissile fuel macroscopic density
11	RSBR5	$\bar{\rho}_{s5}$	Cladding macroscopic density
12	RSBR6	$\bar{\rho}_{s6}$	Subassembly can-wall macroscopic density
13	RSBR7	$\bar{\rho}_{s7}$	Solid control macroscopic density
14	RSBR8	$\bar{\rho}_{s8}$	Intragranular fission gas macroscopic density
15	RSBR9	$\bar{\rho}_{s9}$	Intergranular fission gas macroscopic density
16	TS1	T_{S1}	Fabricated fuel temperature
17	TS2	T_{S2}	Frozen fuel temperature
18	TS3	T_{S3}	Cladding temperature
19	TS4	T_{S4}	Subassembly can-wall temperature

TABLE XXIX (CONT)

<u>Number</u>	<u>Identifier</u>	<u>Plot</u> <u>Symbol</u>	<u>Film</u> <u>Symbol</u>	<u>Mathematical</u> <u>Description</u>
20	TS5	T_{S5}		Solid control temperature
21	RLFBR1	$\bar{\rho}_{L1}$		Liquid fuel macroscopic density
22	RLFBR5	$\bar{\rho}_{L5}$		Solid fuel particle macroscopic density
23	RLBR1	$\bar{\rho}_{L1}$		Liquid fertile fuel macroscopic density
24	RLBR2	$\bar{\rho}_{L2}$		Liquid fissile fuel macroscopic density
25	RLBR3	$\bar{\rho}_{L3}$		Liquid steel macroscopic density
26	RLBR4	$\bar{\rho}_{L4}$		Liquid sodium macroscopic density
27	RLBR5	$\bar{\rho}_{L5}$		Liquid control macroscopic density
28	RLBR6	$\bar{\rho}_{L6}$		Solid fertile fuel particle macroscopic density
29	RLBR7	$\bar{\rho}_{L7}$		Solid fissile fuel particle macroscopic density
30	RLBR8	$\bar{\rho}_{L8}$		Solid steel particle macroscopic density
31	TL1	T_{L1}		Liquid fuel temperature
32	TL2	T_{L2}		Liquid steel temperature
33	TL3	T_{L3}		Liquid sodium temperature
34	TL4	T_{L4}		Liquid control temperature
35	TL5	T_{L5}		Solid fuel particle temperature
36	TL6	T_{L6}		Solid steel particle temperature
37	RGFBR	$\bar{\rho}_{G1}$		Fuel vapor macroscopic density
38	RGBR1	$\bar{\rho}_{g1}$		Fertile fuel vapor macroscopic density
39	RGBR2	$\bar{\rho}_{g2}$		Fissile fuel vapor macroscopic density

TABLE XXIX (CONT)

<u>Number</u>	<u>Identifier</u>	<u>Plot</u>	<u>Film</u>	<u>Mathematical</u> <u>Description</u>
40	RGBR3		$\bar{\rho}_{g3}$	Steel vapor macroscopic density
41	RGBR4		$\bar{\rho}_{g4}$	Sodium vapor macroscopic density
42	RGBR5		$\bar{\rho}_{g5}$	Control vapor macroscopic density
43	RGBR6		$\bar{\rho}_{g6}$	Fission gas macroscopic density
44	TG		T_G	Vapor mixture temperature
45	FUEL		$\bar{\rho}_{fuel}$	Total macroscopic density of all fuel components
46	STEEL		$\bar{\rho}_{steel}$	Total macroscopic density of all steel components
47	SODIUM		$\bar{\rho}_{sodium}$	Total macroscopic density of all sodium components
48	CONTROL		$\bar{\rho}_{control}$	Total macroscopic density of all control components
49	VG		V_G	Axial component of the vapor velocity
50	UG		U_G	Radial component of the vapor velocity
51	VL		V_L	Axial component of the liquid velocity
52	UL		U_L	Radial component of the liquid velocity

the physical mesh. The left, bottom, right, and top edges of the physical mesh are specified as follows.

$$XL = YB = 0.0 \quad ,$$

$$XR = \sum_{i=1}^{\bar{I}} \Delta r_i \quad , \quad \text{and}$$

$$Y_T = \sum_{j=1}^{\bar{J}} \Delta z_j \quad .$$

If the ratio of the width to the height of the physical mesh is greater than the same ratio for the 4020 plotting rectangle ($1022/900 = 1.1356$), the width of the frame determines the scaling, and the 4020 coordinates for the boundaries of the plot are

$$\begin{aligned} \text{FIXL} &= 0 \quad , & \text{for the left boundary,} \\ \text{FIXR} &= 1022 & \text{for the right boundary,} \\ \text{FIYBB} &= 916 & \text{for the bottom boundary, and} \\ \text{FIYT} &= 916 - 1022 \left(\frac{Y_T - Y_B}{X_R} \right) \quad , & \text{for the top boundary.} \end{aligned}$$

If the width-to-height ratio of the physical mesh is less than the same ratio for the 4020 plotting rectangle, the height of the rectangle determines the scaling, and the 4020 boundary coordinates are

$$\begin{aligned} \text{FIXL} &= 511.0 - 450.0 \left(\frac{X_R}{Y_T - Y_B} \right) \quad , \\ \text{FIXR} &= 511.0 + 450.0 \left(\frac{X_R}{Y_T - Y_B} \right) \quad , \\ \text{FIYB} &= 916.0 \quad , \text{ and} \\ \text{FIYT} &= 16.0 \quad . \end{aligned}$$

The two conversion factors for translating from the physical mesh coordinates to the 4020 frame coordinates then are given by

$$\begin{aligned} \text{XCONV} &= (\text{FIXR} - \text{FIXL})/(\text{X}_R - \text{X}_L) \text{ and} \\ \text{XCONV} &= (\text{FIYT} - \text{FIYB})/(\text{Y}_T - \text{Y}_B) \quad . \end{aligned}$$

Each physical mesh coordinate is multiplied by the appropriate conversion factor, this product is added to FIXL or FIYB, and the sum is converted to an integer to locate the 4020 position.

The three-dimensional perspective plots of SIMMER-II variables are obtained by calling the Los Alamos in-house routine PICTUR included in the SIMMER-II source.

G. Restart Files

The restart capability of SIMMER-II allows long calculations to be divided into short jobs. There are several advantages to having such a capability. First, most computer installations schedule smaller jobs first. Second, there is less likelihood of a computer failure for shorter jobs; if there is a failure, not much computing time is lost. Third, short computer runs provide intermediate results that can be examined before continuing the calculation. Finally, restarts can be used to change selected data during a calculation without rerunning the calculation from the beginning. The use of this latter feature and the required input are described in Appendix Q. The restart file structure and usage are described below.

1. Restart File Structure. All restarts are read and written by subroutines REED and RITE using the FORTRAN BUFFER IN and BUFFER OUT instructions in the binary mode. The following 18 sequential records can be on the restart file depending on the update options chosen

Record 1	FWBGN1	Common Block
2	FWBGN2	Common Block
3	TIME	Common Block
4	PSEDIT	Common Block
5	COMINP	Common Block
6	LOCAL	Common Block
7	KINET	Common Block
8	EOS	Common Block
9	S2DIR	Common Block (AEOS OFF)
10	Fluid dynamics	ASSC ARRAY (LCM and LMEM off)
11	Fluid Dynamics	LCM Space
12	Fluid Dynamics	SIMCOM common Block
13	Fluid dynamics	INTCOM Common Block
14	Neutronics	LCM Space (URANUS off)
15	Neutronics	SIMCOM common Block (URANUS off)
16	Neutronics	INTCOM Common Block (URANUS

		off)
17	Densities and temperatures	(URANUS off)
18	Energies	(URANUS off)

2. Restart File Writing and Reading. Break-point, time-limit, and end-of-problem restarts are written depending on input data, system checks, and the option SPECDMP. If the SPECDMP option is off, all restart dumps are written alternately on logical units 21 and 22; odd-numbered restart dumps are written on unit 21 and even-numbered restart dumps are written on unit 22. If the SPECDMP option is on, the restart dumps are written on logical unit 21. If the option FTNLTS is on, each restart dump is written on a separate file (TAPEB, TAPEC,...TAPEZ) for a total of 25 restart dumps per machine run.

The break-point dumps are controlled by T2OMD, the machine time interval dump, and the DTDMP-TDMP tables (for the SPECDMP option on). The time-limit or end of problem restart occurs when the machine time limit or the final problem time (TWFIN) has been reached.

Each restart is flagged by a tape dump number, $N = 1, 2, 3, \dots$, and by IDMP,

where

IDMP = 0, an abnormal end has occurred and no restart file is written;

- = 1, used as an intermediate flag in XCHANJ, EXFLUD, and IMFLUD for error detection (IDMP is later changed to 6);
- = 2, machine time has been exhausted after SHLDXS in GRIND or sense switch 1 has been turned on;
- = 3, $RTIM \geq TWFIN$ in PKDRIV in neutronics;
- = 4, neutronics steady-state dump;
- = 5, T2OMD breakpoint dump from HYDRO, or SPECOMP pump taken;
- = 6, $T \geq TWFIN$, machine time has been exhausted, or sense switch 1 has been turned on in HYDRO;
- = 7, ($NUMIT \geq NITMAX$ and $ITPI \geq 10$) or ($NUMRQ \geq NITMAX$ and $ITRQ10 \geq 10$) in HYDRO;
- = 8, T2OMD breakpoint dump after the neutronics calculation.

To begin a SIMMER-II computer run from a restart, the restart file used as initial conditions for the run must be on logical unit 8 (TAPE8). In addition, the restart number (labeled TAPE DUMP number on the printed output) of the file must match the value of the first word (ITS) on the first problem control card.

H. Postprocessing Features

SIMMER-II writes two files that can be processed to provide plots and tables summarizing the calculation. The first such file is written on logical unit 6 and provides a time history of the space-dependent neutronics parameters and the summations and system balances from the fluid dynamics. The second file, on unit 36, contains mesh cell values of the various fluid dynamics variables at selected points in time.

Extensive postprocessing capabilities are available with the SIMMER-II postprocessor T6P. These capabilities are described in the T6P manual⁵ along with the file contents and format for both the neutronics and fluid dynamics history files (units 6 and 36, respectively).

I. Computing-System-Dependent Features

SIMMER-II has been developed almost exclusively on the CDC-7600 with the Livermore Time-Sharing System (LTSS). However, Los Alamos use of SIMMER-II in recent years has been almost exclusively on a Cray-1 using the CTSS operating system. SIMMER-II also has been tested on the Digital Equipment Corporation VAX 11/780 and DEC-10 computers, the IBM 3033 and 370/168 computers, the AMDAHL 470V, the FACOM-M200, and the CDC-7600 using the SCOPE operating system. From experience gained in the development and exportation of SIMMER-II, features for these systems have been included in the SIMMER-II source but may not have been tested completely. Because computers with 32-bit memories require double-precision variables for the SIMMER-II calculations, each constant used by SIMMER-II is stored in a variable that is defined to be double precision by the DBL option. These variables are listed in Appendix K.

1. LTSS FTNLTS System. FTNLTS is the LTSS system with a FORTRAN compiler (FTN) that has standard FORTRAN as a subset. The FTNLTS version of SIMMER-II is obtained by defining the FTNLTS and the FTN42 option for the SIMMER-II preprocessor. The SCOPE, IBM, and CRAY options are not defined. For FTNLTS systems, the files used in SIMMER-II are defined on the program card by TAPFn, where n is between 1 and 99. File spacing on FTNLTS is done by the subroutine SKIP (NUNIT,

NREC). Logical unit 63 is used for ENCODE and DCODE. The sequence of instructions are

```
REWIND 63
WRITE (63, F1) list
REWIND 63 and
READ (63, F2) list.
```

The namelist instruction for FTNLTS is READ(n,blnm), where blnm refers to the block name of the namelist data.

Other peculiarities of FTNLTS are as follows.

1. LOCF(A) gives the location of A.
2. Large-core memory is defined by LEVEL 2, list of variable names; all LCM variables must be listed on a LEVEL 2 card.
3. In the IF (UNIT,N) test, UNIT is a logical function and the statement is written IF (UNIT(N)) N₁, N₂, N₃.
4. In the IF (EOF,N) test, EOF is a logical function and the statement is written IF (EOF(N)) N₁, N₂.
5. For random I/O, FTNLTS uses the subroutines RDISK and WDISK using a word address relative to 0 for each file.

2. CDC-7600 SCOPE System. SCOPE is a batch system with a FORTRAN compiler (FTN) that has standard FORTRAN as a subset. The SCOPE version of SIMMER-II is obtained by defining the SCOPE and the FTN42 options for the SIMMER-II preprocessor; the FTNLTS, IBM, and CRAY options are not defined. For SCOPE systems, the files used in SIMMER-II are defined on the program card by TAPEn, where n is between 1 and 99. TAPEn has the same meaning as TAPEn on FTNLTS. File spacing on SCOPE is done by the instruction BACKSPACE n and READ(n). Logical unit 63 is used as with FTNLTS for ENCODE and DECODE. SCOPE differs from FTNLTS in the following two ways.

1. Each overlay begins with a program card. Only the program card for the 0,0 overlay has the list of the file definitions. The program cards are preceded by the card OVERLAY (fn, P,S), where fn is the file name where the overlay will reside and P,S is the overlay number;
2. For random I/O, SCOPE uses a table-lookup method to find the disk address instead of a relative section pointer.

3. IBM Operating System. An attempt has been made to include all known IBM system dependencies. To obtain a version of SIMMER-II for IBM systems, the IBM and the DBL option are defined in the SIMMER preprocessor. The FTNLTS,

SCOPE, FTN42, and CRAY options are not defined. The option DBL should be defined because there are a different number of bits for single precision between IBM and the CDC-7600. DBL has to be defined for comparison of problems run on IBM and CDC-7600 computers.

4. Digital Equipment Corporation (DEC) Operating Systems. The DEC operating systems are similar to the IBM systems. To obtain a version of SIMMER-II for the DEC systems, the IBM, DEC, and DBL options are defined. The FTNLTS, SCOPE, FTN42, and CRAY options are not defined. Again DBL should be defined.

5. System-Dependent Routines. SIMMER-II relies on several system-dependent subroutines. Some system-dependent subroutines have been programmed and included in the SIMMER-II source under the different system options. See Appendix S for a complete list of the system-dependent subroutines.

REFERENCES

1. "Update Reference Manual," Control DATA Corporation Publication No. 60342500 (updated).
2. Opcode, Inc., "Historian Plus Users Manual," (April 1983).
3. P. Henrici, Elements of Numerical Analysis, (John Wiley and Sons, New York, 1964), pp. 91-92.
4. K. D. Lathrop and F. W. Brinkley, "TWOTRAN-II: An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport," Los Alamos Scientific Laboratory report LA-4848-MS (July 1973).
5. F. R. Parker, "SIMMER-II Postprocessor Manual", Los Alamos National Laboratory report NUREG/CR-4396, LA-1054a-M (October 1985).

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions to SIMMER-II from many sources, too numerous to mention in detail.

Contributors from Los Alamos include L. L. Smith and C. R. Bell, both of whom have provided many of SIMMER-II's models and methods as well as direction and oversight to the program. F. R. Parker's contributions to SIMMER-II's programming and code management, as well as the neutronics preprocessor and the postprocessor were invaluable to the success of SIMMER-II. R. Steinke and R. Smith made significant contributions to the fluid-dynamics methods and phenomenological models, respectively. G. DeVault support SIMMER-II development in a variety of ways. Finally, the word processing abilities of C. Gonzales and S. Rabener and the editorial abilities of M. Timmers and J. Holmes are appreciated.

The contributions of S. Kleinheins from KfK are appreciated, especially his impressive effort to document the SIMMER-II fluid dynamics programming. The contributions of M. Ishikawa of PNC to the cross-section methods and documentation also are significant.

Finally, the nurturing and guidance of R. Curtis and C. Kelber from the US Nuclear Regulatory Commission since the beginning (1974) in support of the SIMMER program is greatly appreciated. The efforts of J. Boudreau and M. Stevenson in pioneering the SIMMER program were invaluable.

APPENDIX A

NOMENCLATURE

<u>Symbol</u>	<u>Definition</u>
A	Area (m^2) used to compute the thermal response time of frozen material.
A	Input coefficient (m/s) for the liquid fuel sonic velocity.
$\bar{A}$	Average molecular weight.
a	Subscript to indicate conditions just below the interface in the multicomponent vaporization-condensation model.
A_{clad}	Transient fuel pin cladding surface area per mesh cell unit volume (m^{-1}).
A_{cLmLk}	Contact area (m^2) of droplet of energy component m with a droplet of energy component k.
A_{flow}	Cross-sectional flow area per unit volume (m^{-1}).
A'_{flow}	Flow channel cross-sectional area per unit volume (m^{-1}) before including frozen material.
A_{Gsk}	Structure surface area (m^{-1}) of component k in contact with the vapor field in the vaporization-condensation model.
$A_{h,d}$	Factor used to update the delayed neutron precursor concentration for neutronics mesh cell ij and delayed family d from the delayed neutron source at a previous time.

A_i	Surface area per unit volume (m^{-1}) of interface i . Can also be used with a k subscript.
$A_{i\pm 1/2j}$	Cell surface areas on the radial cell boundaries of the neutronics mesh. Used in the transport theory neutronics solutions.
A_k	Coefficients of a cubic polynomial in the rational function approximation.
A_{Lm}	Surface area per unit volume (m^{-1}) of liquid-field energy component m .
A_{LmSk}	Structure surface area per unit volume (m^{-1}) of structure-field energy component k seen by liquid-field energy component m .
A_M	Molecular weight of material M .
A_{pell}	Transient pellet surface area per unit volume (m^{-1}).
A_{pin}	Pin surface area per unit volume (m^{-1}) after frozen material is considered.
A'_{pin}	Pin surface area per mesh cell unit volume (m^{-1}) before considering frozen material.
A_{Sk}	Surface area per unit volume (m^{-1}) of structure-field energy component k .
A_{Smk}	Heat flow area per unit volume (m^{-1}) between structure-field energy components m and k .
A_{tot}	Total surface area per unit volume (m^{-1}) for the pin and can-wall structure after frozen material is included.

A'_{tot}	Total surface area per unit volume (m^{-1}) for the pin and can-wall structure before frozen material is included.
A_{ursf}	Cross-sectional area (m^{-1}) of unrestructured fuel.
$a_{G,M}$	Equation-of-state input quantity (K) for material M used to increase the gas constant when the vapor temperature is above the material's critical temperature.
$a_{L,M}$	Equation-of-state parameter (J/kg) used to obtain continuity in the specific energy for a liquid at two-thirds of a material's critical temperature.
a_m	Dummy variable (J/m^3) used to solve the vapor energy equation in the vaporization-condensation model.
a_{M10} a_{M20} a_{M30} a_{M11} a_{M21} a_{M31}	Fitting constants used internally by SIMMER-II to obtain the functional relationship of the gas constant on the local pressure. Determined from conditions given on page (III-38).
a_o	A constant denoting the amount of full donor-cell weighting to be used.
a_{1m} a_{2m} a_{3m} a_{4m}	Input parameters for the temperature-dependent liquid microscopic density used in Eq. (III-71) below two-thirds of the critical temperature.
A^{-1}	Part of the iteration matrix used for solution of the rebalance equation used for the S_n transport equation solution or solution of the neutron diffusion equation. [$A^{-1}(L+U)$ is the complete matrix.]
A_{can}^i	Initial can-wall surface area per mesh cell unit volume (m^{-1}).

A_{clad}^i	Initial fuel pin cladding surface area per mesh cell unit volume (m^{-1}).
A_{pell}^i	Initial fuel pellet surface area per mesh cell unit volume (m^{-1}).
A_{pin}^i	Initial pin surface area per unit volume (m^{-1}).
$\underline{A}_F^{n+1}$	Coefficient matrix for the implicit solution of the mass continuity equations.
a_F^{n+1}	Elements of the matrix $\underline{A}_F^{n+1}$ possessing double i and j subscripts defined by Eqs. (IV-220) through (IV-224).
$\underline{A}_G^{n+1}$	Coefficient matrix, $\underline{A}_F^{n+1}$, specialized only to the vapor field.
$a_{gij,ij}^{n+1}$	Diagonal matrix element set to unity for the vapor field in single-phase cells.
$a_{1,m}^{\kappa}, a_{2,m}^{\kappa}$	Parameters calculated for iteration κ as defined by Eqs. (IV-98) and (IV-99) for the vaporization-condensation model.
A,B,C	Symbolic variables used in Eq. (IV-114) to represent a quadratic equation for the vapor temperature.
ALDRG	Input parameter giving the structure volume fraction above which structure drag is infinite.
ALPMIN	Input variable giving the minimum liquid volume fraction permitted before K_{GL} modifications occur. Steel particles added if $\alpha_L < 0.01$ ALPMIN.
B	Input exponent for the liquid fuel sonic velocity expression.

b	Subscript to indicate conditions just above the interface in the multicomponent vaporization-condensation model.
b(T)	Second coefficient (m^3/kg) in the equation-of-state expansion used in the SESAME library.
$B_{h,d}$	Factor used to update the delayed neutron precursor concentration for neutronics mesh cell ij and delayed family d from the delayed neutron source at a previous time.
B_i	Cell surface areas on the axial cell boundaries of the neutronics mesh. Used in the transport theory neutronics solutions.
B_λ, B_θ	Temporary variables used in the pressure iteration. Defined by Eqs. (IV-197) and (IV-198).
b_m	Dummy variable ($\text{J}/\text{m}^3\cdot\text{K}$) used to solve the vapor energy equation in the vaporization-condensation model.
b_o	A constant denoting the amount of linearly interpolated convective fluxing to be used.
$b_{1m} \ b_{2m} \ b_{3m}$	Input parameters for the temperature-dependent liquid microscopic density used in Eq. (III-71) above two-thirds of the critical temperature.
C	Input minimum (m/s) for the liquid fuel sonic velocity expression.
C	Proportionality constant. Product of C_1 and C_2 in the liquid-liquid heat transfer formalism.

c	Sound speed (m/s) for a multicomponent, multiphase fluid.
$c(T)$	Third coefficient (m^6/kg^2) in the equation-of-state expansion used in the SESAME library.
C_D	Form drag coefficient.
$C_{GL1}, C_{GL2}, C_{GL3}, C_{GL4}$	Input coefficients for the liquid-field/vapor-field heat-transfer correlation.
$C_{GS1}, C_{GS2}, C_{GS3}, C_{GS4}$	Input coefficients for the structure-field/vapor-field heat-transfer correlation.
C_{ijd}	Delayed neutron precursor concentration for precursor family d averaged over neutronics mesh cell ij (at time t).
$C_{LS1m}, C_{LS2m}, C_{LS3m}, C_{LS4m}$	Input coefficients for the structure-field/liquid-field heat-transfer correlation.
C_ℓ	Chebyshev polynomial of degree ℓ .
$C_{o,ijd}$	Delayed neutron precursor concentration for precursor family d averaged over neutronics mesh cell ij (at a previous time step).
C_{ORF}	Radially dependent orifice coefficients.
C_t	Courant input constant. Fraction of the velocity Courant condition to be used as a time step control.
C_1	Proportionality constant between the contact area in droplet-droplet collisions and the smaller droplet cross-sectional area.

C_1 and C_2	Functions giving the residual between a derivative and the lower and upper difference approximations in the rational function approximation.
C_2	Fraction of the residence time that should characterize the average collision between two liquid drops.
C_λ, C_θ	Temporary variables used in the pressure iteration. Defined by Eqs. (IV-199) and (IV-200).
c_d	Adjoint-flux-weighted space and energy integral of $\chi_d C_d$. The delayed neutron precursor concentration used in the amplitude equation.
c_L	Liquid sonic velocity (m/s) for a single component system.
c_{Lm}	Sonic velocity (m/s) for liquid-field energy component m. The cell location can be indicated by i and j indices.
$c_{\ell m}$	Sonic velocity (m/s) for liquid-field density component m.
c_p	Constant-pressure volume specific heat (J/kg·K) of the vapor mixture. Used in the multicomponent vaporization-condensation model.
c_p	Heat capacity (J/kg·K) used to compute the thermal response time of frozen material.
$\bar{c}_{pG}$	Average constant-pressure specific heat (J/kg·K) of the vapor mixture. Greek κ superscript can be used to indicate an iteration number.

c_{pGM}	Constant-pressure specific heat (J/kg·K) of the vapor phase of material M.
$\bar{c}_{vG}$	Average constant-volume specific heat (J/kg·K) of the vapor mixture. Greek κ superscripts can be used to indicate an iteration counter.
c_{vGM}	Constant-volume specific heat (J/kg·K) of the vapor phase of material M.
c_{vLM}	Constant-volume specific heat (J/kg·K) for the liquid phase of material M.
c'_{vLM}	Constant-volume specific heat (J/kg·K) averaged between the liquid and vapor energy curves above two-thirds of a material's critical temperature.
c_{vm}	Constant-volume heat capacity of energy component m.
c_{vSM}	Constant-volume specific heat (J/kg·K) for the solid phase of material M.
ϵ^2	Square of the effective sonic velocity (m ² /s ²) for the liquid mixture.
COAL	Coalescence coefficient for determining the particle droplet size. Defined by Eq. (C-7).
$\tilde{D}$	Mass continuity equation residual (kg/m ³ ·s). Can refer to either the liquid or vapor field. Can be superscripted with a κ to indicate the pressure iteration number. Can be subscripted with i and j to indicate the cell location. Can be underlined to indicate a vector comprising all ($\bar{i} \bar{j}$) cells.

$\underline{D}$	Symbolic middle block diagonal matrix forming part of the decomposition of $\underline{M}$ and accounting for the spatial coupling of a line of mesh cells. A j subscript indicates the mesh cell line number.
D_h	Hydraulic diameter (m) of the flow channel.
D_{ijg}	Neutron diffusion coefficient for neutronics mesh cell ij in group g .
$\underline{D}_j^*$	Tridiagonal submatrix for mesh cell line j . Defined by Eq. (G-15) for $j = 1$ and by Eq. (G-19) for $j > 1$.
$(\underline{D}_j^*)^{-1}$	Inverse of the $\underline{D}_j^*$ matrix.
D_M	Equation of state derivative of R_M , defined by Eq. (IV-8).
D_M'	Derivative of R_M defined following Eq. (IV-40). In particular, see Eq. (IV-41).
D_m	Mass diffusivity (m^2/s) of material m in the boundary layer used in the multicomponent vaporization-condensation model.
$(DW)_{ij}$	Weight function for integration of Doppler reactivity for fluid-dynamics mesh cell ij . (Used for the Uranus option.)
D_λ, D_θ	Temporary denominators used in the pressure iteration. Defined by Eqs. (IV-201) and (IV-202).
d_m	Dummy variable (J/kg) used to solve the vapor-energy equation in the vaporization-condensation model.

D_h^i	Channel hydraulic diameter (m) before considering frozen material.
D_{hc}^i	Hydraulic diameter (m) for intact geometry.
D_{hcan}^i	Hydraulic diameter (m) for the subassembly with no fuel pins.
D_{hf}^i	Hydraulic diameter (m) for fuel pellets with no cladding.
$\tilde{D}_G^k$	The total mass continuity equation error ($\text{kg}/\text{m}^3 \cdot \text{s}$) for the vapor field in mesh cell (i,j).
$\tilde{D}_L^k$	The total mass-continuity equation error ($\text{kg}/\text{m}^3 \cdot \text{s}$) for the liquid field in mesh cell (i,j).
$\frac{dT_{S1}}{dr}$	Local temperature gradient in the fuel.
$\frac{dT_{S1}}{dt}$	The transient heating rate in the fuel.
$e_{cM}(v_M), p_{cM}(v_m)$	"Cold curve" energy and pressure at zero kelvin for material M.
$e_{Con,M}$	Condensate energy (J/kg) for the liquid phase of material M. κ superscripts can be used to indicate the iteration number.
$e_{Crt,M}$	The specific internal energy (J/kg) for material M at its critical temperature.
e_{Fm}	Breakup energy (J/kg) of structure-field energy component m.

e_G	Specific internal energy (J/kg) of the vapor mixture. K superscripts can be used to indicate the iteration number, the cell location can be indicated by i and j indices.
$\tilde{e}_G$	Intermediate value of specific vapor internal energy (J/kg) to be used for determining the initial pressure in starting the pressure iteration.
$e_{G,M}$	Specific internal energy (J/kg) of vapor material M.
$e_{G,M}^*$	Parameter (J/kg) used in the analytic equation of state and defined by Eq. (III-47).
$e_{Liq,M}$	Liquidus energy (J/kg) of material M that is assigned to structure-field energy component m.
e_{Lm}	Specific internal energy (J/kg) of liquid-field energy component m. The cell location can be indicated by i and j indices.
e_M	Specific energy (J/kg) of material M.
e_m	Specific energy (J/kg) of energy component m.
e_{Sm}	Specific internal energy (J/kg) of structure-field energy component m. The cell location can be indicated by i and j indices.
$e_{Sol,M}$	Solidus energy (J/kg) of material M that is assigned to structure-field energy component m.
$e_{Vap,M}$	Specific vapor saturation energy (J/kg) for material M.
e_G^n	Current (beginning of time step) specific internal energy (J/kg) for the vapor mixture.

e_G^{n+1}	Advanced-time specific internal energy (J/kg) for the vapor mixture.
$\tilde{e}_G^{n+1}$	Estimated vapor mixture specific internal energy (J/kg) at the estimated advanced time $\tilde{t}^{n+1}$.
e_{Lm}^n	Current (beginning of time step) specific internal energy (J/kg) for liquid-field energy component m.
e_{Lm}^{n+1}	Advanced-time specific internal energy (J/kg) for liquid energy component m.
$\tilde{e}_{Lm}^{n+1}$	Estimated advanced-time specific internal energy (J/kg) for liquid-field energy component m.
$\tilde{e}_{Sk}^{n+1}$	Estimated advanced-time specific internal energy (J/kg) for structure-field energy component k.
e_{Sm}^n	Current (beginning of time step) specific internal energy (J/kg) for structure-field energy component m. Additional i and j subscripts can be used to identify a cell location.
e_{Sm}^{n+1}	Advanced-time specific internal energy (J/kg) for structure energy component m.
$e_{Vap,M}^{n+1}$	Advanced-time specific vapor saturation energy (J/kg) for material M.
F	Symbolic representation for a function. When used in the text, the function referred to should be obvious. Also can be written as lower case and subscripted with i and j to denote a particular cell.
F	Total effective (integrated over space and energy with adjoint flux weighting) neutron source.

$\underline{F}$	An unknown vector to be determined by the matrix inversion methods described in Appendix G. Can possess a k or numeric superscript to indicate an iteration number. An intermediate value is superscripted by $\hat{k}$.
f	Friction factor.
F_{gp}	Dimensionless input parameter used to simulate fuel flow in intersubassembly gaps.
$FD_{ij\pm 1/2}$	Neutron flow across axial boundaries in the negative axial direction of neutronics mesh cell ij (of group g at time t). neutron source at a previous time.
$FL_{i\pm 1/2j}$	Neutron flow across radial boundaries in the negative radial direction of neutronics mesh ij (of group g at time t).
$FR_{i\pm 1/2j}$	Neutron flow across radial boundaries in the positive radial direction of neutronics mesh cell ij (of group g at time t).
$FU_{ij\pm 1/2}$	Neutron flow across axial boundaries in the positive axial direction of neutronics mesh cell ij (of group g at time t).
f_{Fm}	Input fraction of the heat of fusion through which the component energy is increased before breakup of the component.
f_g	Correction factor (dimensionless) for energy and mass-transfer rates.
f_{ij}	Flux-shape correction factor, used to enforce neutron conservation in the S_n transport solution.

f_{Lmin}	Minimum single-phase liquid-structure friction factor.
f_{Lm}	Melt fraction of liquid-field energy component m.
f_{Lm}	Melt fraction of liquid-field density component m. neutron source at a previous time.
f_{LS1}, f_{LS2}	Input constants used in the single-phase liquid-structure friction factor correlation.
f_M	Empirical parameter (J/kg·K) defined to equal the gas constant for material M below the critical temperature.
f_p	Fission gas release adjustment factor to account for the effect of the pressure in the fuel.
f_{rgr}	Fission gas release adjustment factor to account for the effect of fuel grain size.
f_{Sm}	Melt fraction of structure-field energy component m.
f_{sm}	Melt fraction of structure-field density component m.
f_{ursf}	Fraction of the fabricated fuel that is unrestructured.
f_1	Time step reduction function used in the quasistatic method.
f_2	Time step reduction function used in the quasistatic method.
$\underline{f}^n$	The n^{th} iterate of the vector containing rebalance factors for the S_n transport equation solution or fluxes for the diffusion equation solution. neutron source at a previous time.

$F_{ij,d}^d$	Spatial delayed-neutron shape source term for neutronics mesh cell ij and delayed family d .
$F_{c,ijg}^M$	Shielding factor for the capture reaction of material M in neutronics mesh cell ij in group g .
$F_{d,ijg}^M$	Shielding factor for the elastic downscatter reaction from group g to $g+1$ of material M in neutronics mesh cell ij .
$F_{e,ijg}^M$	Shielding factor for the elastic scattering reaction of material M in neutronics mesh cell ij in group g .
$F_{f,ijg}^M$	Shielding factor for the fission cross section of material M in neutronics mesh cell ij in group g .
$F_{t,ijg}^M$	Shielding factor for the total cross section of material M in neutronics mesh cell ij in group g .
$F_{x,ijg}^M$	Shielding factor for the cross section of reaction type x in material M in neutronics mesh cell ij in group g .
$f_{f,ijg}^i$	Shielding factor for the fission reaction of isotope i in neutronics mesh cell ij and energy group g .
$f_{t,ijg}^i$	Shielding factor for the total cross section of isotope i in neutronics mesh cell ij and energy group g . neutron source at a previous time.
$f_{x,ijg}^i$	Shielding factor for reaction type x of isotope i in neutronics mesh cell ij and energy group g .
$f_{c,ijg}^k$	Shielding factor for the capture reaction of isotope k in neutronics mesh cell ij for group g .

$f_{d,ijg}^k$	Shielding factor for the elastic downscatter reaction of isotope k in neutronics mesh cell ij in group g . (Used only for NEUPREP-on methods.)
$f_{e,ijg}^k$	Shielding factor for the elastic scattering reaction of isotope k in neutronics mesh ij in group g . (Used in both in-group and downscatter calculations if NEUPREP is off. Used only for in-group elastic scatter methods if NEUPREP is on.)
f_{Lm}^{n+1}	Advanced-time value of the melt fraction of liquid-field energy component m .
f_{Sm}^{n+1}	Advanced-time value of the melt fraction of structure-field energy component m .
f_{sm}^{n+1}	Advanced-time value of the melt fraction of structure-field density component m .
f_{Lm}^{n+1}	Advanced-time value of melt fraction of liquid-field density component m . neutron source at a previous time.
$f_{\rho_{s8}}^0$	Fission gas release adjustment factor to account for the effect of the initial fission gas concentration.
$f\left(\frac{dT_{S1}}{dr}\right)$	Fission gas release adjustment factor to account for the effect of the temperature gradient in the fuel.
$f\left(\frac{dT_{S1}}{dt}\right)$	Fission gas release adjustment factor to account for the effect of the fuel heating rate.
FAIL3	Variable indicating whether motion and/or intermesh heat transfer is allowed at the right-hand boundary of a cell. Possible values are given in Table VII.

g	Mass transfer coefficient ($\text{kg}/\text{m}^2 \cdot \text{s}$) obtained by the unit Lewis number assumption (so it is not species dependent) and corrected for a finite mass transfer rate.
g^*	Mass transfer coefficient ($\text{kg}/\text{m}^2 \cdot \text{s}$) in the limit of small mass transfer rates.
$\vec{g}$	Acceleration of gravity. Without the vector, the axial component is understood.
G_G	Inverse of the vapor field compressibility (m^2/s^2 or $\text{m}^3 \cdot \text{Pa}/\text{kg}$).
G_L	Updated value for the inverse of the liquid field compressibility (m^2/s^2 or $\text{m}^3 \cdot \text{Pa}/\text{kg}$). neutron source at a previous time.
$\tilde{G}_{Lij}$	Estimate of the inverse of the liquid-field compressibility (m^2/s^2 or $(\text{Pa} \cdot \text{m}^3)/\text{kg}$) for cell (i,j) given by Eq. (IV-145). Also may be used without the ij subscripts.
g_M	Equation-of-state parameter (dimensionless) defined by Eq. (III-55) and used to fit the gas constant for material M when the vapor temperature exceeds the critical temperature in the limit of small mass transfer rates.
$G_{Lo}^{-1} = \frac{1}{G_{Lo}}$	Liquid compressibility ($\text{kg}/\text{m}^3 \cdot \text{Pa}$) extrapolated to the pure liquid state.
h	Heat-transfer coefficient ($\text{W}/\text{m}^2 \cdot \text{K}$) used to compute the thermal response time of frozen material.

h^*	Heat-transfer coefficient ($W/m^2 \cdot K$) in the limit of zero mass transfer.
$H_{eff,mk}$	Product of the heat transfer coefficient and surface area ($W/m^3 \cdot K$) actually used for energy transfer between energy components m and k . Defined following Eq. (IV-67).
H_{GLm}	Product of the vapor-side heat-transfer coefficient and interfacial area ($W/m^3 \cdot K$) for liquid-field energy component m , as corrected for mass transfer. κ superscripts or numeric superscripts can be used to indicate an iteration number. neutron source at a previous time.
H_{GLm}^*	Product of the vapor-side heat-transfer coefficient and interfacial area ($W/m^3 \cdot K$) for liquid-field energy component m , uncorrected for mass transfer.
H_{Gsk}	Product of the vapor-side heat transfer coefficient and the vaporization-condensation surface area ($W/m^3 \cdot K$) for structure-field energy component k , as corrected for mass transfer. κ superscripts can be used to indicate an iteration number.
$\hat{H}_{Gsk}$	Product of the surface area and the effective heat transfer coefficient ($W/m^3 \cdot K$) for direct structure-to-vapor heat transfer.
H_i	Decay heat rate for the decay heat group i .
H_{LGm}	Product of the liquid-side heat transfer coefficient and interfacial area ($W/m^3 \cdot K$) for liquid-field energy component m at a droplet interface. Possibly modified by Eq. (IV-89).

H_{Lm}	Product of the structure heat transfer area and the liquid-side heat transfer coefficient ($W/m^3 \cdot K$) for liquid energy component m.
H_{SGk}	Product of the vaporization-condensation structure surface area and the structure-side heat transfer coefficient ($W/m^3 \cdot K$) for structure energy component k. neutron source at a previous time.
H_{Sk}	Product of the structure surface area and the structure-side heat transfer coefficient ($W/m^3 \cdot K$) for structure energy component k.
h_{can}	Transient heat-transfer coefficient ($W/m^2 \cdot K$) for the subassembly can wall.
h_{clad}	Transient cladding heat-transfer coefficient ($W/m^2 \cdot K$).
$h_{f,M}$	Heat of fusion (J/kg) of material M corresponding to energy component m.
h_{froz}	Heat-transfer coefficient ($W/m^2 \cdot K$) for frozen material.
h_{GLm}	Vapor-side heat-transfer coefficient ($W/m^2 \cdot K$) for liquid-field energy component m.
h_{GS}	Vapor-side heat-transfer coefficient ($W/m^2 \cdot K$) for vapor-to-structure heat transfer.
h_k	Vapor-side heat-transfer coefficient ($W/m^2 \cdot K$) for surface k.
h_{LGm}	Liquid-side heat-transfer coefficient ($W/m^2 \cdot K$) for liquid-field energy component m at a droplet surface.

$h_{lg,M}$	Heat of vaporization (J/kg) for material M. κ or k superscripts can be used to indicate an iteration number. neutron source at a previous time.
$h_{lg,M}^*$	Heat of vaporization (J/kg) for material M when the saturation temperature for material M is zero.
h_{Lm}	Heat-transfer coefficient ($W/m^2 \cdot K$) for liquid-field energy component m. Used for the multicomponent vaporization-condensation model.
h_{LSm}	Liquid-side heat-transfer coefficient ($W/m^2 \cdot K$) of liquid-field energy component m at a structure surface.
h_{ma}	Enthalpy (J/kg) of material m below an interface.
h_{mb}	Enthalpy (J/kg) of material m above an interface.
h_{mk}	Product of overall heat-transfer coefficient and area per unit volume ($W/m^3 \cdot K$) between energy components m and k.
h_o	Heat-transfer coefficient ($W/m^2 \cdot K$) corresponding to bulk conditions in the liquid.
h_{pell}	Transient heat-transfer coefficient ($W/m^2 \cdot K$) for the fuel pellet.
h_{Sk}	Structure-side heat-transfer coefficient ($W/m^2 \cdot K$) for structure-field energy component k.
h_{∞}	Corrected heat-transfer coefficient ($W/m^2 \cdot K$) for mass transfer corresponding to bulk conditions for vapor. neutron source at a previous time.

h_{∞}^*	Heat-transfer coefficient ($\text{W}/\text{m}^2 \cdot \text{K}$) when mass transfer does not occur.
H_{LGm}^0	Unmodified product of the liquid-side heat-transfer coefficient and interfacial area ($\text{W}/\text{m}^3 \cdot \text{K}$) for liquid-field energy component m at a droplet interface.
h_{can}^i	Initial heat-transfer coefficient ($\text{W}/\text{m}^2 \cdot \text{K}$) for the subassembly can wall.
h_{clad}^i	Input cladding heat-transfer coefficient ($\text{W}/\text{m}^2 \cdot \text{K}$).
h_{pell}^i	Initial heat-transfer coefficient ($\text{W}/\text{m}^2 \cdot \text{K}$) for the fuel pellet.
$(h_{\text{g},M}^{\hat{\kappa}+1})_{\text{reset}}$	Recalculation value of the last iterate for the heat of vaporization (J/kg) using Eq. (E-10) when convergence difficulties occur in the vaporization-condensation model.
$h_{\text{g},M}^+$	Heat of vaporization (J/kg) at the liquid temperature of material M .
I	Number of elements in a "line" for use in the AIM matrix solution method. Can be either $\bar{I}$ or $\bar{J}$ depending on orientation. Indicates how far off the main diagonal elements the $\underline{L}$ and $\underline{U}$ matrices are. neutron source at a previous time.
$\bar{I}$	IBAR, the total number of radial (horizontal) mesh intervals.
i	The radial mesh index.
ICW	Can-wall set-over flag. Defined in Sec. IV.C.1.

IGD	Number of delayed neutron precursor groups.
I_{GEOM}	Index specifying either plane geometry ($I_{\text{GEOM}} = 0$) or cylindrical geometry ($I_{\text{GEOM}} = 1$).
I^{r}	Extrapolated leakage contribution to the reactivity for reactivity time step t^{r} .
I^{s}	Leakage contribution to the reactivity for flux shape time step t^{s} .
J	Number of "lines" for use in the AIM matrix solution method. See the definition of I.
j	The axial mesh index.
$\bar{j}$	A line index in Appendix G. Note that $(I \cdot J) = (\bar{I} \cdot \bar{J})$.
$\bar{J}$	JBAR, the total number of axial mesh intervals.
$\vec{j}_{\text{L}}$	Liquid superficial velocity (m/s). neutron source at a previous time.
$ \vec{j}_{\text{L}} $	Magnitude of the liquid superficial velocity.
K	Symbolic representation for a momentum exchange function ($\text{kg/m}^3 \cdot \text{s}$). Spatially subscripted in Eqs. (IV-181) and (IV-182) to show how averaging is performed across momentum cells.
$(\text{KV})_1$	Expression defined by Eq. (IV-63) used to compute the droplet radius by the Weber number condition.
$(\text{KV})_2$	Expression defined by Eq. (IV-64) used to compute the droplet radius by the Weber number condition.

K_{GL}^n	Value of the momentum-exchange function ($\text{kg}/\text{m}^3 \cdot \text{s}$) computed using values from the current time step to model the drag forces between the liquid and vapor fields. neutron source at a previous time.
K_{GL}^{n-1}	Value of the momentum-exchange function ($\text{kg}/\text{m}^3 \cdot \text{s}$) used on the previous time step to model the drag forces between the liquid and vapor fields.
k_{Gm}^{mono}	Thermal conductivity ($\text{W}/\text{m} \cdot \text{K}$) for a monotonic gas of vapor material m.
k_{Gm}^{poly}	Thermal conductivity ($\text{W}/\text{m} \cdot \text{K}$) for a polyatomic gas of vapor material m.
KCHAN	Indicator for which frozen material to expect first in the structure configuration model. KCHAN = 1 for frozen fuel; KCHAN = 2 for frozen control.
KCHPS	Logic parameter in the structure configuration model. When KCHPS = 1, the frozen material volume fractions are calculated. When KCHPS = 2, the frozen material heat-transfer coefficient is determined.
$\underline{L}$	Symbolic lower block diagonal matrix forming part of the decomposition of $\underline{M}$. Can be divided into submatrices by subscripting with j, where j is defined following Eq. (G-11).
$\underline{M}$	Symbolic coefficient matrix used to describe matrix inversion methods in Appendix G. neutron source at a previous time.
$M = M(m)$	Equation-of-state material number for structure-or liquid-field energy component m, or the equation-of-state identification for vapor material component m. neutron source at a previous time.

K_{GL}^n	Value of the momentum-exchange function ($\text{kg/m}^3 \cdot \text{s}$) computed using values from the current time step to model the drag forces between the liquid and vapor fields. neutron source at a previous time.
K_{GL}^{n-1}	Value of the momentum-exchange function ($\text{kg/m}^3 \cdot \text{s}$) used on the previous time step to model the drag forces between the liquid and vapor fields.
k_{Gm}^{mono}	Thermal conductivity ($\text{W/m} \cdot \text{K}$) for a monotonic gas of vapor material m.
k_{Gm}^{poly}	Thermal conductivity ($\text{W/m} \cdot \text{K}$) for a polyatomic gas of vapor material m.
KCHAN	Indicator for which frozen material to expect first in the structure configuration model. KCHAN = 1 for frozen fuel; KCHAN = 2 for frozen control.
KCHPS	Logic parameter in the structure configuration model. When KCHPS = 1, the frozen material volume fractions are calculated. When KCHPS = 2, the frozen material heat-transfer coefficient is determined.
$\underline{L}$	Symbolic lower block diagonal matrix forming part of the decomposition of $\underline{M}$. Can be divided into submatrices by subscripting with j, where j is defined following Eq. (G-11).
$\underline{M}$	Symbolic coefficient matrix used to describe matrix inversion methods in Appendix G. neutron source at a previous time.
$M = M(m)$	Equation-of-state material number for structure-or liquid-field energy component m, or the equation-of-state identification for vapor material component m. neutron source at a previous time.

$(\underline{M}^{\text{approx}})^{-1}$	Symbolic approximate inverse of matrix $\underline{M}$.
N	Amplitude function used in the quasi-static method to solve for the time dependence of the neutron flux. The flux equals the product of the amplitude and the flux shape.
N	Number of mass-transfer rates at each surface in the multicomponent vaporization-condensation model.
N_m	Droplet number density per unit fluid volume (m^{-3}) for liquid energy component k.
Nu_{Lm}	Nusselt number for liquid-field energy component m.
n	The number of the most recent fluid-dynamics time step.
n_k	Droplet number density per unit cell volume (m^{-3}) for liquid energy component m.
n^r	Number of the fluid-dynamics time step that began the current reactivity time step.
$\hat{N}_{ij}^i$	Atom number density of isotope i for neutronics mesh cell ij.
N_{ij}^{im}	Atom number density of isotope i in component m for neutronics mesh cell ij.
$\hat{N}_{ij}^M$	Atom number density of material M for neutronics mesh cell ij.
NCAT	Indicator of the energy component comprising a can wall's surface.

NCGR = 6	Number of density components in the vapor field.
NCHT	Indicator of the energy component comprising a pin's surface.
NCLE = 6	Number of liquid-field energy components.
NCLR = 8	Number of density components in the liquid field.
NCSE = 5	Number of energy components in the structure field.
NCSR = 9	Number of density components in the structure field.
NMAT = 5	Number of materials.
NNMAT = 5	Number of material mixtures used to compute the neutron flux. In SIMMER-II these are the five material components: fertile fuel, fissile fuel, stainless steel, sodium, and control.
NTR	Mesh cell indicator for the geometric configuration.
p	Local field pressure (Pa).
p	Local pressure (Pa) in the fuel.
$\bar{p}$	Intermediate value of the mesh cell pressure (Pa) to be used as an estimate in the pressure iteration. A κ can be used for the iteration number and subscripts can be added for cell indices.
$\hat{p}_{VM(M)}$	Effective vapor pressure (Pa) for material M.
Pr _G	Vapor mixture Prandtl number.

Pr_{Lm}	Prandtl number for liquid-field energy component m.
$p_c(\rho)$	Pressure (N/m^2) as a function of density at zero degrees kelvin.
$P_{Crt,M}$	Critical pressure (Pa) for material M.
P_{flow}	Wetted perimeter (m) of the flow channel.
P_G	Pressure (Pa) resulting from vapor in a one-component system.
P_{GM} or P_{Gm}	Partial pressure (Pa) of vapor material M. Also can be equivalently subscripted by a lower case m.
P_i	Total pressure (Pa) from volatile materials at an interface.
P_i	Pressure (Pa) at the inner radius of the unrestructured fuel.
$P_i, P_{i+1}, P_{i-1},$ $P_j, P_{j+1}, P_{j-1},$	Pressure (Pa) for cells (i,j), (i+1,j), (i-1,j), (i,j), (i,j+1), and (i,j-1), respectively.
P_L	Pressure (Pa) resulting from liquid in a single component system.
$\hat{p}_L$	Sum of the liquid-field vapor pressures (Pa).
P_{Lm}	Single-phase pressure (Pa) in liquid-field energy-component m.
$\hat{p}_{Lm}$	Effective vapor pressure (Pa) for liquid energy component m. Equivalent to $\hat{p}_{vM(m)}$.

$P_M(v_M, c_M)$	Pressure (Pa) of material M.
P_M^* and T_M^*	Input parameters for the vapor pressure curve for material M. For a single-component system, these can be written without the M subscript.
P_{nf}	Pressure (Pa) in the volume fraction not available for flow.
P_o	Pressure (Pa) at the outer radius of the unrestructured fuel.
$P_{T,m}$	Minimum pressure (Pa) that liquid component m can exert either by compressing other liquids or by vaporizing.
$P_v(T_L)$	Vapor pressure (Pa) at the given liquid temperature in a single component system. Also can be written without the liquid temperature indicated.
$P_v(T_{sat})$	Vapor pressure (Pa) in a single-component system that is consistent with the vapor pressure curve evaluated at T_{sat} .
$P_{vM}(T_{Lm})$	Liquid vapor pressure (Pa) for liquid-field energy component m evaluated at the component temperature. If material and energy components are identical, they can be replaced by an identical subscript, for example, k.
$P_{vM}(\tilde{T}_{Sat,M})$	Vapor pressure (Pa) for material M evaluated at a temperature of $\tilde{T}_{Sat,M}$.
P_y	Yield pressure (Pa) because of circumferential stress.
P_1	Explicit term (m/s) used in the droplet radius evaluation associated with the Weber number condition.

p^i	Input vapor-field pressure (Pa).
p^n	Current (beginning of time step) cell pressure (Pa).
p^{n+1}	Advanced-time pressure (Pa). Can be subscripted with i and j variables to indicate cell location and have difference operators Δ_λ and Δ_θ applied to indicate gradients.
p^o	Pressure (Pa) for a single-phase cell as determined by the input processor.
$\tilde{p}_{est}^{n+2}$	Estimated pressure (Pa) at the end of time step $n+2$, inferred from Eq. (IV-209).
Q	Known right-side vector in the generalized matrix equation [Eq. (G-3)], solved in Appendix G.
Q_G	Total energy-transfer rate per unit volume (W/m^3) to the vapor mixture.
Q_{ij}	Energy source distribution for the Uranus option. (Power per unit mass or power per unit volume.)
Q_{Lm}	Total energy-transfer rate per unit volume (W/m^3) to liquid-field energy component m .
Q_{mij}	Specific power (W/kg) of component m for fluid-dynamics mesh cell ij . (Used only for the Uranus option.)
Q_{Ngm}	Normalized internal energy-generation rate per unit mass (W/kg) of density component m .
Q_{NLm}	Normalized internal energy-generation rate per unit volume (W/m^3) for liquid-field energy component m .

$Q_{N\ell k}$	Normalized internal energy-generation rate per unit mass (W/kg) for liquid-field density component k.
$Q_{Nmi j}$	Specific power (W/kg) of component m for neutronics mesh cell ij.
Q_{Nsk}	Normalized internal energy-generation rate per unit mass (W/kg) for structure-field density component k.
Q_{NSm}	Normalized internal energy-generation rate per unit volume (W/m ³) for structure-field energy component m. Additional i and j subscripts can be added to indicate a cell location.
Q_{Sm}	Total energy-transfer rate per unit volume (W/m ³) to structure-field energy component m.
q_{cLmLk}	Heat transfer rate per collision (W) between a droplet of liquid energy component m and one of liquid energy component k.
q_{HG}	Total heat-transfer rate (W/m ³) to the vapor field.
q_{HLL1}	Total heat-transfer rate per unit volume (W/m ³) from liquid fuel to the other liquid field components.
q_{HLL2}	Total heat-transfer rate (W/m ³) from liquid steel to the other liquid-field components.
q_{HLm}	Total heat-transfer rate (W/m ³) to liquid-field energy component m.
q_{HLmLk}	Heat-transfer rate per unit volume (W/m ³) from liquid-field energy component k to liquid-field energy component m.

$(q_{HLmLk})_{\max}$	Maximum heat-transfer rate per unit volume (W/m^3) in the limit of small droplet radii in transferring heat from liquid-field energy component k to liquid-field energy component m.
q_{HLmSk}	Heat-transfer rate per unit volume (W/m^3) from structure-field energy component k to liquid-field energy component m.
q_{Hm}	Total heat-transfer rate (W/m^3) to energy component m.
$\tilde{q}_{Hm}$	Intermediate total heat transfer rate per unit volume (W/m^3) to energy component m. This term includes the contribution from implicit heat transfer of Sec. IV.C.6.
q_{HSCL1}	Total heat-transfer rate (W/m^3) from liquid fuel to the structure-field components.
q_{HSm}	Total heat-transfer rate (W/m^3) to structure-field energy component m.
q_{HSmLk}	Heat-transfer rate per unit volume (W/m^3) from liquid-field energy component k to structure-field energy component m.
q_{HSmSk}	Heat-transfer rate per unit volume (W/m^3) from structure-field energy component k to structure-field energy component m.
q_{HSPL1}	Total heat-transfer rate (W/m^3) from liquid fuel to the pin structure components.
q_{HS3L2}	Heat-transfer rate (W/m^3) from liquid steel to the pin structure.

$\tilde{q}_{H,S4}$	Intermediate can-wall heat flux (W/m^3) as computed by Eq. (IV-74), to be used in the can-wall set-over model.
$\tilde{q}_{H,S4,i+1,j}$	Initialization can-wall heat flux ($W/m^3 \cdot K$) in cell (i+1,j) resulting from the can-wall setover contribution. Also accomodates the drag heating contribution.
$\tilde{q}_{H,S4,i-1,j}$	Intermediate can-wall heat flux (W/m^3) in cell (i-1,j) after addition of the contribution resulting from can-wall setover.
$\tilde{q}'_{H,S4,i-1,j}$	Intermediate can-wall heat flux (W/m^3) in cell (i-1,j) before addition of the contribution resulting from can-wall setover.
q_{HS4L2}	Heat-transfer rate (W/m^3) from liquid steel to the can-wall structure.
q_{IGmLm}	Energy transfer rate per unit volume (W/m^3) to the vapor field from the liquid-vapor interface of liquid component m. κ superscripts can be used to indicate the iteration number.
q_{IGmSk}	Energy transfer rate per unit volume (W/m^3) from the interface of structure-field energy component k to the vapor field by the vaporization-condensation process. κ superscripts can be used to indicate the iteration number.
q_{ILmGm}	Energy transferred per unit volume (W/m^3) from the interface of liquid-field energy component m into the liquid component during vaporization or condensation. κ superscripts can be used to indicate the iteration number.

q_{ILmSk}	Heat flow per unit volume (W/m^3) to liquid energy component m from the surface of structure component k.
q_{ISk}	Net heat flow rate per unit volume (W/m^3) at the surface of structure energy component k. A negative value implies melting (heat flow to the liquid), and a positive value implies freezing (heat flow to the structure).
q_{ISkGm}	Energy transfer rate per unit volume (W/m^3) from the interface of structure-field energy component k into the structure component from the vaporization-condensation process. κ superscripts can be used to indicate the iteration number.
q_{ISkL}	Heat flow per unit volume (W/m^3) from the surface of structure energy-component k into the structure interior from liquid-to-structure energy transfer.
q_{KG}	Total energy transfer rate to the vapor field (W/m^3) because of drag heating.
q_{KGL}	Total viscous heating (W/m^3) because of drag between the liquid field and the vapor field.
q_{KGS}	Total viscous heating (W/m^3) because of drag between the structure field and the vapor field.
q_{KLm}	Total energy transfer rate per unit volume (W/m^3) because of drag heating to liquid component m.
q_{KLS}	Total viscous heating (W/m^3) because of drag between the structure field and the liquid field.
q_{KSm}	Total energy transfer rate (W/m^3) to structure-field energy component m because of drag heating.

q_{mk}	Heat transfer rate (W/m^3) from energy component k to energy component m.
q_{vG}	Total energy transfer rate (W/m^3) to the vapor field because of viscous heating.
q_{vLm}	Energy generation rate per unit volume (W/m^3) in liquid-field energy component m because of viscous heating when the option to model shear flow between adjacent cells is active.
q_{xi}, q_{yj}	Functions to interpolate x and y based on the nearest tabulated points.
$\tilde{q}_{H,S4,i+1,j}^t$	Can-wall heat flux ($W/m^3 \cdot K$) to cell (i+1,j) resulting from can-wall setover before normalization.
$\tilde{q}_{H,S4,i-1,j}^t$	Can-wall heat flux ($W/m^3 \cdot K$) to cell (i-1,j) resulting from can-wall setover before normalization.
q_{rG}	Total energy transfer rate (W/m^3) to the vapor field resulting from mass transfer.
q_{rGmLm}	Energy transfer rate per unit volume (W/m^3) to vapor material component m from liquid energy component m resulting from vaporization (positive) or condensation (negative).
q_{rLm}	Total energy transfer rate per unit volume (W/m^3) resulting from mass transfer to liquid field energy component m.
q_{rLmF}	Energy transfer rate per unit volume (W/m^3) taken from liquid energy component m associated with mass transfer resulting from freezing.

$q_{\Gamma LmF}$	Energy transfer per unit volume (W/m^3) for liquid-field energy component m resulting from freezing but only including nonequilibrium effects.
$q_{\Gamma LmGm}$	Energy transfer rate per unit volume (W/m^3) to liquid-field energy component m from vapor-field material component m resulting from vaporization (negative) or condensation (positive).
$q_{\Gamma LmM}$	Energy transfer rate per unit volume (W/m^3) appearing in liquid energy component m because of melting.
$q_{\Gamma SkF}$	Energy transfer rate per unit volume (W/m^3) appearing in structure component k associated with the mass transfer resulting from freezing.
$q_{\Gamma SkM}$	Energy transfer rate per unit volume (W/m^3) taken from structure energy component k due to melting. This quantity has a negative sign.
$q_{\Gamma Sm}$	Total energy transfer rate (W/m^3) as a result of mass transfer to structure-field energy component m.
$q_{\Gamma SmLk}$	Energy-transfer rate (W/m^3) from structure-field energy component m to liquid-field solid energy component k as a result of solid mass transfer.
$q_{\Gamma SmLk}$	Energy-transfer rate (W/m^3) from structure-field energy component m to liquid-field energy component k as a result of liquid mass transfer.
R	Universal gas constant ($J/kg \cdot K$)
R	Dimensionless input variable used to apportion structure drag between liquid and vapor. Also used for the liquid-to-structure heat transfer contact area.

r	Radial coordinate (m).
$R_{\text{Crt},M}$	Gas constant (J/kg·K) for material M given by the critical pressure divided by the product of the critical density and critical temperature.
R_M	Gas constant (J/kg·K) for material M.
$\tilde{R}_M$	Intermediate gas constant (J/kg·K) for material M used in treating single-phase cells.
R_{Mi}	Gas constant (J/kg·K) for material M at infinite dilution.
R_{mk}	Multiplier on liquid-liquid heat transfer between energy components m and k.
R_{Mo}	Gas constant (J/kg·K) for material M when both the saturation temperature for material M and the vapor temperature are 95% of the critical temperature for material M.
Re_G	Vapor-field Reynolds number.
Re_{GLm}	Reynolds number used for vapor flow around a sphere of liquid-field energy component m.
Re_{Lm}	Reynolds number for liquid-field energy component m.
r_c	Droplet radius (m) of the component with the smaller sized droplet in liquid-liquid heat transfer.
r_{gr}	Fuel grain radius (m) in the unrestructured fuel.
r_i	Inner radius (m) of the unrestructured fuel.

$r_{i\pm 1/2}$	Radial coordinate of the boundary of a neutronics mesh cell with radial index i .
$r_{i-1}, r_{i-1/2}, r_i, r_{i+1/2}, r_{i+1}$	Radius (m) of the fluid-dynamics mesh at locations $i-1, i-1/2, i, i+1/2$, and $i+1$.
r_o	Outer radius (m) of the unrestructured fuel.
r_p	Radius (m) for a symbolic liquid drop.
r_{pm}	Droplet radius (m) of liquid-field energy component m .
$r_{p,max}$	Maximum allowed liquid droplet radius (m).
$r_{p,min}$	Minimum allowed liquid droplet radius (m).
$r_{pm,1}, r_{pm,2}, r_{pm,3}, r_{pm,4}, r_{pm,5}$	Trial droplet radii (m) for liquid-energy component m computed using various criteria as discussed in Sec. III.D.3 and following Eq. (IV-65).
R_M^{n+1}	Advanced-time gas constant (J/kg·K) for material M .
r_{clad}^i	Outer radius (m) of the fuel pin cladding.
r_{pell}^i	Outer radius (m) of the fuel pellet.
r_{pm}^n	Current (beginning of time step) droplet radius (m) for liquid energy component m .
r_{pm}^{n+1}	Advanced-time droplet radius for liquid energy component m .
ROGCUT	Input variable used for cutoff tests that concern both the vapor density and the product of the vapor density and temperature.

RSCLAD	Macroscopic cladding input density (kg/m^3) below which failure occurs.
S	Number of surfaces in a mesh cell in the multicomponent vaporization-condensation model.
s	Entropy ($\text{J/kg}\cdot\text{K}$) of the vapor mixture.
S_i	Difference quotient for the rational function approximation defined by Eq. (IV-262).
S_{ijg}	Total neutron source to energy group g for neutronics mesh cell ij.
T	Absolute temperature (K).
t	Time (s).
T_{Cont}	Mass-averaged temperature of $\bar{p}_{\text{Cont}}$ (in neutronics mesh cell ij at time t).
$(T_{\text{Cont}})_{ij}$	Mass-averaged temperature of $(\bar{p}_{\text{Cont}})_{ij}$ (at time t).
$T_{\text{Crt},M}$	Critical temperature (K) for material M.
T_{eff}	Temperature (K) actually used to transfer energy to a component in the implicit heat transfer calculation, Sec. IV.C.6.
T_{Fert}	Mass-averaged temperature of $\bar{p}_{\text{Fert}}$ (in neutronics mesh cell ij at time t).
$(\bar{T}_{\text{Fert}})_{ij}$	Mass-averaged temperature of $(\bar{p}_{\text{Fert}})_{ij}$ (at time t).
T_{Fiss}	Mass-averaged temperature of $\bar{p}_{\text{Fiss}}$ (in neutronics mesh cell ij at time t).

$(\bar{T}_{Fiss})_{ij}$	Mass-averaged temperature of $(\bar{\rho}_{Fiss})_{ij}$ (at time t).
T_G	Vapor mixture temperature (K). κ superscripts or numeric subscripts can be used to indicate an iteration number or, in the case of the multicomponent vaporization-condensation model, the value for a short time step of length δt^κ .
$\tilde{T}_G$	Intermediate value of the vapor temperature (K) used for treating single-phase cells. Also used in the vaporization-condensation for representing direct structure to vapor heat transfer.
T_i	Surface temperature (K) for the non-equilibrium vaporization-condensation model. May be superscripted by "old" and "new" identifiers for use in a relaxation procedure.
T_{ij}	Flux shape "tilt" for neutronics mesh cell ij.
T_{ki}	Interface temperature (K) of surface k.
T_L	Liquid temperature (K) for a hypothetical case where there is only one liquid component.
T_{Lm}	Temperature (K) of liquid-field energy component m. κ superscripts indicate the value at the beginning of a short vaporization-condensation time step.
$\hat{T}_{Lm}$	Effective temperature (K) for liquid-field energy component m used for the vaporization-condensation model. Defined by Eqs. (IV-90) through (IV-92).
$(T_{Lm})_i$	Interface temperature (K) of liquid component m.

$T_{\text{Melt,Fuel}}$	Melting temperature (K) of fuel.
$T_{\text{Melt,M}}$	Melting temperature (K) of material M.
$T_{\text{Melt,Steel}}$	Melting temperature (K) of steel.
$T_M(v_M, c_M)$	Temperature (K) of material M.
T_{Na}	Mass-averaged temperature of $\bar{p}_{\text{Na}}$ (in neutronics mesh cell ij at time t).
$(\bar{T}_{\text{Na}})_{ij}$	Mass-averaged temperature of $(\bar{p}_{\text{Na}})_{ij}$ (at time t).
T_o	Temperature (K) for bulk material below a vaporization-condensation interface.
T_s	A structure surface temperature (K).
T_{sat}	Saturation temperature (K) in a single-component system. Gives consistency between the vapor-pressure curve and the actual vapor species partial pressure.
$T_{\text{Sat,M}}$	Saturation temperature (K) for the partial pressure because of the vapor species of the same composition as liquid-field energy component m. κ or $\hat{\kappa}$ superscripts can be used to indicate an iteration number.
$\tilde{T}_{\text{Sat,M}}$	Intermediate value of the saturation temperature used in treating single-phase cells.
T_{Sm}	Temperature (K) of structure-field energy component m. κ superscripts indicate the value at the beginning of a short vaporization-condensation time step.
T_{Smi}	Interfacial temperature (K) of solid energy component m.

T_{SS}	Mass-averaged temperature of $\bar{p}_{SS}$ (in neutronics mesh cell ij at time t).
$(\bar{T}_{SS})_{ij}$	Mass-averaged temperature of $(\bar{p}_{SS})_{ij}$ (at time t).
$T_{Sup,M}$	Input superheat (K) for material M.
T_{∞}	Free stream vapor temperature (K) above a vaporization-condensation interface.
t_{cLmLk}	Contact time (s) of a droplet of energy component m with a droplet of energy component k.
t_{froz}	Thickness (m) of frozen control material or frozen fuel material.
t_i	Functional derivative in the rational function approximation, defined by Eq. (IV-262) and approximated by Eq. (IV-263).
t^r	Time of the most recent reactivity time step.
t^s	Time of the most recent flux-shape time step.
$\tilde{t}^{n+1}$	Estimated time (s) using the time step estimate $\tilde{\Delta t}$ obtained before the exchange-function calculation.
$\bar{T}_{ij}^i$	Average temperature of isotope i for neutronics mesh cell ij.
T_{S1}^i	Minimum fuel temperature (K) for intragranular fission gas release.
$\bar{T}_{ij}^m$	Average temperature of component m for neutronics mesh cell ij.

T_G^n	Current (beginning-of-time-step value) for the vapor mixture temperature (K).
T_G^{n+1}	Advanced-time value of the vapor mixture temperature (K).
T_{Lm}^n	Current (beginning-of-time-step value) temperature (K) for liquid-field energy component m.
T_{Lm}^{n+1}	Advanced-time temperature (K) of liquid-field energy component m.
$\tilde{T}_{Lm}^{n+1}$	Estimated advanced-time temperature (K) for liquid energy component m.
$\tilde{T}_m^{n+1}$	Estimated advanced time temperature (K) for structure or liquid energy component m. Additional i and j subscripts can be added to indicate a cell location.
$T_{Sat,M}^n$	Current (beginning of time step) saturation temperature (K) corresponding to the vapor partial pressure of material M.
$T_{Sat,M}^{n+1}$	Advanced time saturation temperature (K) corresponding to the vapor partial pressure of material M.
$\tilde{T}_{Sk}^{n+1}$	Estimated advanced time temperature (K) for structure-field energy component k. Additional i and j subscripts can be added to indicate a cell location.
T_{Sm}^n	Current (beginning of time step) temperature (K) for structure-field energy component m.
$(\tilde{T}_{Cont}^0)_{ij}$	Mass-averaged temperature of $(\bar{p}_{Cont})_{ij}$ (at time $t = 0$).

$(\bar{T}_{\text{Fert}}^0)_{ij}$	Mass-averaged temperature of $(\bar{\rho}_{\text{Fert}})_{ij}$ (at time $t = 0$).
$(\bar{T}_{\text{Fiss}}^0)_{ij}$	Mass-averaged temperature of $(\bar{\rho}_{\text{Fiss}})_{ij}$ (at time $t = 0$).
$(\bar{T}_{\text{Na}}^0)_{ij}$	Mass-averaged temperature at $(\bar{\rho}_{\text{Na}})_{ij}$ (at time $t = 0$).
$(\bar{T}_{\text{SS}}^0)_{ij}$	Mass-averaged temperature of $(\bar{\rho}_{\text{SS}})_{ij}$ (at time $t = 0$).
T_{Lm}^0	Beginning iterate for the temperature (K) of liquid-field energy component m.
t_{can}^i	Thickness (m) of the subassembly can wall.
T_m^n	Beginning of time step temperature (K) for energy component m.
$\underline{U}$	Symbolic upper block diagonal matrix forming part of the decomposition of $\underline{M}$. Can be given a line index subscript.
$\underline{U}^*$	Collapsed tridiagonal upper block matrix defined by Eq. (G-18) and the associated discussion. Can be subscripted by j to indicate a submatrix block. The quantity j is defined following Eq. (G-11).
U_{cLmLk}	Overall heat-transfer coefficient ($\text{W/m}^2 \cdot \text{K}$) following contact of a droplet of energy component m with a droplet of energy component k.
$U_{\text{Gi}+1/2}, U_{\text{Li}+1/2},$ $U_{\text{Gi}-1/2}, U_{\text{Li}-1/2}$	Radial velocities (m/s) evaluated for the vapor (G) and liquid (L) fields at locations given by the indices $(i+1/2, j)$ and $(i-1/2, j)$. Also can written with the j subscript included.

$(\tilde{U}_G)_\lambda, (\tilde{U}_L)_\lambda$	Intermediate vapor- and liquid-field radial velocities (m/s) at the $(i+1/2, j)$ interface. A κ superscript indicates the iteration number for the pressure iteration.
U_{ij}	Radial velocity (m/s) for either the liquid or vapor field at cell location (i, j) . Can be written with different i and j indices, including λ to represent $i+1/2$ and θ to represent $j+1/2$.
U_{Li}, U_{Gi}	Liquid and vapor velocities (m/s) averaged over interfaces $(i+1/2, j)$ and $(i-1/2, j)$. If used with a difference operator, Δ_j , the evaluation is done as illustrated by Eq. (IV-137).
U_F^{n+1}	Advanced-time radial velocities (m/s) for either the liquid (l) or vapor (g) fields. Used with i and j subscripts to indicate the cell boundary considered.
$U_{G\lambda}^{n+1}, U_{L\lambda}^{n+1}$	Radial velocities (m/s) at the advanced time for the vapor and liquid fields evaluated at cell location $(i+1/2, j)$.
$\bar{V}$	Average relative velocity (m/s) during a collision between two liquid droplets.
$ \vec{V} $	Magnitude of either the liquid or vapor velocity (m/s). Defined in terms of axial, V , and radial, U , components by Eq. (IV-124).
v	Specific volume (m^3/kg) of a mixture of components.
$\vec{V}_G$	Vapor-field velocity vector with components U_G (m/s) and V_G (m/s) in the r and z directions, respectively. Velocity components may be subscripted with both i and j variables to indicate the cell interface.

$ \vec{v}_G $	Magnitude of vapor-field velocity (m/s).
v_g	Average velocity of neutron within energy group g (m/s).
$v_{Gj+1/2}, v_{Lj+1/2}$ $v_{Gj-1/2}, v_{Lj-1/2}$	Axial velocities (m/s) evaluated for the vapor (G) and liquid (L) fields at locations given by the indices $(i, j+1/2)$ and $(i, j-1/2)$. Also can be written with both subscripts included.
$ \vec{v}_{GL} $	Absolute velocity difference (m/s) between the liquid field and the vapor field.
$(\tilde{v}_G)_\theta, (\tilde{v}_L)_\theta$	Intermediate vapor- and liquid-field axial velocities (m/s) at the $(i, j+1/2)$ interface. A κ superscript indicates the iteration number for the pressure iteration.
v_i	Volume (m^3) of mesh cell i, j with the j subscript omitted.
v_{ij}	Axial velocity (m/s) for either the liquid or vapor field at cell location (i, j) . Can be written with different spatial indices, including λ to represent $i+1/2$ and θ to represent $j+1/2$. written as $v_{i,j}$.
v_{ij}	Volume of cells in the neutronics mesh. (Fluid-dynamics cells may be subdivided into multiple neutronics cells.)
v_{i+1}	Volume (m^3) of mesh cell $i+1, j$ with the j subscript omitted. Also can appear with the j subscript.
v_{i-1}	Volume (m^3) of mesh cell $i-1, j$ with j subscript omitted. Also can appear with the j subscript.

$\vec{v}_L$	Liquid-field velocity vector, with components U_L (m/s) and V_L (m/s) in the directions r and z , respectively. Velocity components may be subscripted with both i and j variables to indicate the cell location.
v_{ij}	Volume (m^3) of mesh cell i, j . Also can be
$ \vec{v}_L $	Magnitude of the liquid-field velocity (m/s).
v_L	Specific volume (m^3/kg) in a single-component system.
$\vec{v}_{LG}$	Relative velocity between the liquid field and the vapor field.
$\bar{v}_m$	Average random velocity (m/s) for liquid droplets of energy component m .
v_M	Specific volume (m^3/kg) of material M .
v_{ma}	Velocity component (m/s) of material m below and perpendicular to an interface.
v_{mb}	Velocity component (m/s) of material m above and perpendicular to an interface.
VOL_m	Volume swept per unit time and volume ($m^{-3} \cdot s^{-1}$) by all droplets or particles of liquid energy component m .
VOL_{s_m}	Volume (m^3) swept by a droplet of radius r_{pm} in time Δt .
v_F^{n+1}	Advanced-time axial velocities (m/s) for either the liquid (l) or vapor (g) fields. Used with i and j subscripts to indicate the cell boundary considered.

$ V_{GL}^n $	Velocity difference (m/s) between the liquid and vapor fields at the beginning of a time step.
$ V_{GL}^{n+1} $	Estimated velocity difference (m/s) between the liquid and vapor fields at the end of a time step.
$v_{G\theta}^{n+1}, v_{L\theta}^{n+1}$	Axial velocities (m/s) at the advanced time for the vapor and liquid fields evaluated at cell location (i,j+1/2).
$V(E)$	Neutron velocity (m/s).
VIS	Vapor-field viscous heating term (W/m^3). Evaluated by Eq. (IV-166).
W	Work term (W/m^3) in the vapor energy equation, evaluated at the beginning of a time step.
$\hat{W}$	Work term (W/m^3) in the vapor energy equation, evaluated at the end of a time step.
$\underline{W}$	Diagonal element relaxation matrix used in the AIM method described in Appendix G.
$(W_{Cont})_{ij}$	Control fuel reactivity worth ($\delta k/kg$) for fluid dynamics mesh cell i,j. (Used only for the Uranus option.)
We	Critical Weber number input by the user for determining liquid droplet radii.
W_{eff}	Effective work term (W/m^3) combining W and $\hat{W}$ as given in Eq. (IV-226). When underlined, $\underline{W}_{eff}$ can represent a vector of elements $\bar{I} \cdot \bar{J}$ in length.

W_{ex}	Energy loss rate per unit volume (W/m^3) resulting from fission gas crossing a two-phase/single-phase interface.
$(W_{Fert})_{ij}$	Fertile fuel reactivity worth ($\delta k/kg$) for fluid dynamics mesh cell i,j . (Used only for the Uranus option.)
$(W_{Fiss})_{ij}$	Fissile fuel reactivity ($\delta k/kg$) for fluid dynamics mesh cell i,j . (Used only for the Uranus option.)
$(W_{Na})_{ij}$	Sodium fuel reactivity worth ($\delta k/kg$) for fluid dynamics mesh cell i,j . (Used only for the Uranus option.)
$(W_{SS})_{ij}$	Steel fuel reactivity worth ($\delta k/kg$) for fluid dynamics mesh cell ij . (Used only for the Uranus option.)
w_{ll}	Elements of the diagonal relaxation matrix $\underline{W}$.
w_M	Mole fraction of material M .
w_m	Weight assigned to angular interval m for angular quadratures in the angular flux solution of the S_n neutron conservation equation.
We_m	Weber number for a particular of liquid energy component m .
X	Numerator of expression (J/m^3) used to calculate the vapor temperature in the multicomponent vaporization-condensation model.
x	Transverse coordinate (m) used in planar geometry.
x	Coordinate in the direction of the liquid-vapor velocity difference used in Eq. (IV-52).

x	Dependent iterative variable ($W/m^3 \cdot K$) in the expression to define the mass-transfer effect on the vapor-side heat-transfer coefficient. A κ superscript can be used to indicate the iteration number.
x_{Fk}	Fraction of structure-field energy component k transferred to the liquid momentum-field as liquid.
$\tilde{x}_{gm}$	Intermediate value for vapor-field density component m mass fractions.
x_{gmG1}	Mass fraction of vapor density component m in vapor energy component 1 (fuel vapor).
$x_{\ell mLk}$	Mass fraction of liquid density component m in liquid energy component k .
$\tilde{x}_{\ell mL1}$	Mass fraction of liquid density component m in liquid energy component 1 using mass fractions evaluated from $\tilde{\rho}_{Lm}^n$.
x_m	Mass fraction of vapor material m in the vapor mixture.
x_m	Parameter to indicate the presence of liquid component m for further vaporization. If $x_m = 1$, the nominal case exists. If $x_m = 0$, all the liquid of component m vaporizes.
x_{mi}	Mass fraction of component m at interface i .
$x_{p\infty}$	Mass fraction of vapor material m in the vapor free stream.
x_{smSk}	Mass fraction of structure density component m in structure energy component k .

$x_1, x_2, \dots, x_N$	Tabulated dependent variables used to explain the rational function interpolation algorithm.
Y	Denominator of expression ($J/m^3 \cdot K$) used to calculate the vapor temperature in the multicomponent vaporization-condensation model.
y	Distance (m) from an interface where vaporization/condensation can take place.
$y_1, y_2, \dots, y_N$	Second tabulated dependent variable used in the explanation of the rational function interpolation algorithm.
$y^{K-2}, y^{K-1}, y^K, y^{K+1}$	Various iterates of a quantity, y , where acceleration to convergence is to be attempted by Steffensen's method as given in Eq. (IV-116).
z	Axial coordinate (m).
$z_{j\pm 1/2}$	Axial coordinate of the boundary of a neutronics mesh cell with axial index j .
$z_{j-1}, z_{j-1/2}, z_j$	Axial problem dimension (m) at the height of mesh location
$z_{j+1/2}, z_{j+1}$	$j-1, j-1/2, j, j+1/2$, and $j+1$.
z_{LmLk}	Total collision rate per unit time and per unit cell volume of liquid-energy component m with liquid-energy component k .
z_m	Frequency of droplet collisions per unit time and fluid volume ($s^{-1} \cdot m^{-3}$) for liquid-energy component m .
$z_{m \rightarrow k}$	Collision rate per unit fluid volume ($m^{-3} \cdot s^{-1}$) of component m with component k resulting from movement of component m .

α	Relaxation parameter used in the multicomponent vaporization-condensation model.
α	Thermal diffusivity (m^2/s) used in the multicomponent vaporization-condensation model.
α	Weight factor used in combining boundary angular flux shapes to yield a cell average flux shape (for neutronics mesh cell i,j ; angular interval m ; and energy group g at time t).
α_{can}	Can-wall transient volume fraction.
α_{clad}	Transient cladding volume fraction.
α_{froz}	Volume fraction of frozen material.
α_G	Vapor-field volume fraction. The cell location can be indicated by i and j indices.
α_{Go}	Vapor volume fraction at the transition point between single-phase and two-phase flow.
α_{GSR}	Fraction of the flow channel that must be vapor to initiate vapor field contact with structure surfaces.
α_{ijmg}	Correction term to the numerator of the angular flux shape for the weighted-diamond S_n method for neutronics mesh cell i,j ; angular interval m ; and energy group g at time t .
α_L	Total volume fraction of the liquid-field components. The cell location can be indicated by i and j indices.

α_f	Coefficient of the current fission source in the linear combination of previous fission sources to Chebyshev accelerate the outer iteration of the neutronics conservation equation solution.
α_{Lm}	Volume fraction of liquid-field energy component m.
α_{Lm}	Volume fraction of liquid-field density component m.
α_{Lmo}	Volume fraction of liquid-field energy component m at the transition point between single- and two-phase flow.
α_{Lo}	Liquid-field volume fraction at the transition point between single- and two-phase flow.
$\alpha_{m\pm 1/2}$	Coefficients in the angular redistribution term of the S_n neutron conservation equation.
α_{nf}	Volume fraction not available for flow based on the presence of cladding.
$\alpha_{nf,cw}$	Volume fraction not available for flow based on the presence of can wall.
α_o	The fraction of the available flow volume determining the transition between two-phase and single-phase flow.
α_p	Particle volume fraction.
α_p	Pin structure volume fraction.
α'_p	Pin volume fraction when the frozen material is not included.

α_{pell}	Transient pellet volume fraction.
α_{S}	Total volume fraction of the structure field. The cell location can be indicated by i and j indices.
α_{sm}	Volume fraction of structure-field density component m.
α_{t}	Thermal diffusivity (m^2/s) used in the liquid-liquid heat transfer model.
$\alpha_{\text{C}}^{\text{i}}$	Energy deposition per neutron capture in isotope i.
$\alpha_{\text{can}}^{\text{i}}$	Initial volume fraction of the subassembly can wall.
$\alpha_{\text{clad}}^{\text{i}}$	Initial volume fraction of the cladding.
$\alpha_{\text{f}}^{\text{i}}$	Energy deposition per fission in isotope i.
$\alpha_{\text{G}}^{\text{i}}$	Initial vapor volume fraction as specified by user input.
$\alpha_{\text{L}}^{\text{i}}$	Total liquid-field volume fraction as specified by user input.
$\alpha_{\text{Lm}}^{\text{i}}$	Initial volume fraction for liquid-field energy component m as input by the user.
$\alpha_{\text{Lm}}^{\text{i}}$	Initial volume fraction for liquid-field density component m as input by the user.
$\alpha_{\text{pell}}^{\text{i}}$	Initial volume fraction of the fuel pellet.
$\alpha_{\text{c,g}}^{\text{M}}$	Energy deposition per neutron capture in cross-section material M in energy group g.

$\alpha_{f,g}^M$	Energy deposition per fission in cross-section material M in energy group g.
α_G^n	Current time (beginning of time step) total vapor volume fraction.
α_G^{nf}	Vapor volume fraction ignoring convection during the current time step.
α_G^{n+1}	Advanced-time total vapor volume fraction. Can be subscripted by i and j to indicate a cell location or by λ or θ to indicate values on cell boundaries.
$\tilde{\alpha}_G^{n+1}$	Unadjusted advanced-time total vapor volume fraction used to compute a cell's single-phase pressure.
α_{Go}^{n+1}	Advanced-time vapor volume fraction at the transition between single-phase and two-phase flow.
$\alpha_{G,est}^{n+2}$	Estimated vapor-field volume fraction at the end of time step n+2.
α_L^n	Current time (beginning of time step) total liquid field volume fraction.
α_L^{n+1}	Advanced-time total volume fraction of the liquid-field components. Can be subscripted by i and j to indicate a cell location or by λ or θ to indicate values on cell boundaries.
$\alpha_{L,est}^{n+2}$	Estimated liquid volume fraction at the end of time step n+2, computed with $\bar{\rho}_{L,est}^{n+2}$.
α_S^n	Current time (beginning of time step) total volume fraction of the structure field.

α_S^{n+1}	Advanced-time total volume fraction of the structure field. Can be subscripted by i and j to indicate a cell location.
$\tilde{\alpha}_S^{n+1}$	Advanced-time total volume fraction of the structure field before addition of the nonflow volume fraction contributions.
α_G^0	Initial vapor volume fraction determined by the input processor.
α_L^0	Initial liquid-field volume fraction after adjustment by the input processor.
α_{Lm}^0	Initial volume fraction for liquid-field energy component m after adjustment by the input processor.
β	Total effective delayed neutron fraction.
β_d	Effective delayed neutron fraction for delayed neutron precursor family d.
β_{hi}	Decay heat fraction for the decay heat group i.
β_{ijmg}	Numerator of the angular flux shape in the diamond-difference S_n method for neutronics mesh cell i,j; angular interval m; and energy group g at time t.
β_ℓ	Coefficient of the (current iterate - 2) fission source in the linear combination of previous fission sources to Chebyshev accelerate the outer iteration of the neutron conservation equation solution.
β^r	Total effective delayed neutron fraction for the most recent reactivity step.

β^κ	Coefficient matrix (s/m^2) in iteration κ of the pressure iteration derived in Appendix F from the $\partial\tilde{D}/\partial\tilde{p}$ definition in Eq. (IV-212).
Γ	Sum of the mass-transfer rates per unit volume ($\text{kg/m}^3\cdot\text{s}$) at an interface.
$\bar{\Gamma}$	Symbolic representation for a mass-transfer rate ($\text{kg/m}^3\cdot\text{s}$). Spatially subscripted in Eqs. (IV-183) and (IV-184) to indicate how averaging is performed across momentum cells.
Γ_G	Total mass-transfer rate ($\text{kg/m}^3\cdot\text{s}$) from the vapor field.
Γ_{GL}	Total mass-transfer rate ($\text{kg/m}^3\cdot\text{s}$) from the vapor field to the liquid field. Can have i and j subscripts to indicate a cell location.
Γ_{gm}	Total mass-transfer rate per unit volume ($\text{kg/m}^3\cdot\text{s}$) from vapor-field density component m.
Γ_{GmLk}	Total rate of mass transfer per unit volume ($\text{kg/m}^3\cdot\text{s}$) from vapor-field material component m to liquid-field energy component k.
$\Gamma_{gm\rightarrow k}$	Mass-transfer rate per unit volume ($\text{kg/m}^3\cdot\text{s}$) from vapor density component m to liquid density component k.
$(\Gamma_{GmLm})_{ISk}$	Mass-transfer rate per unit volume ($\text{kg/m}^3\cdot\text{s}$) of vapor-field material component m to liquid-field energy component m resulting from condensation occurring at the interface of structure-field energy component k. κ superscript can be used to indicate an iteration counter.

$\Gamma_{g6,extra}$	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of fission gas to compensate for previously calculated losses across a single-phase/two-phase interfacial boundary.
Γ_{IGmLm}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from vapor-field material component m to liquid-field energy component m due to phase transition occurring at the liquid-component interface. κ superscript can be used to indicate an iteration counter.
Γ_{ILmGm}	Rate of mass transfer per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid-field energy component m to vapor-field material component m due to phase transition occurring at the liquid-component interface. κ superscript can be used to indicate an iteration counter.
Γ_{Intra}	Intragranular fission gas release rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of unrestructured fuel.
Γ_L	Total mass-transfer rate ($\text{kg}/\text{m}^3 \cdot \text{s}$) from the liquid field.
Γ_{LG}	Total mass-transfer rate ($\text{kg}/\text{m}^3 \cdot \text{s}$) from the liquid field to the vapor field. Can have i and j subscripts to indicate a cell location.
Γ_{Lm}	Total mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid-field energy component m.
Γ_{LmF}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) because of freezing of liquid component m.
Γ_{LmGm}	Total rate of mass transfer per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid-field energy component m to vapor-field material component m.

$(\Gamma_{LmGm})_{ISk}$	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of liquid-field energy component m to vapor-field material component m resulting from vaporization occurring at the interface of structure-field energy component k. κ superscript can be used to indicate an iteration counter.
Γ_{LmLk}	Total rate of mass transfer per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid-field energy component m to liquid-field energy component k.
Γ_{LmSk}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid energy component m to structure energy component k.
$\Gamma_{\Delta m}$	Total mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid-field density component m.
$\Gamma_{\Delta m, \text{extra}}$	Additional mass transfer ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid field density component m from vapor convected across a two-phase/single-phase cell interface.
$\Gamma_{\Delta mgk}$	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid density component m to vapor density component k.
$\Gamma_{\Delta mLk}$	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid-field density component m to liquid-field density component k.
$\Gamma_{\Delta mSk}$	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid-field density component m to structure-field density component k.
Γ_{LS}	Total mass-transfer rate ($\text{kg}/\text{m}^3 \cdot \text{s}$) from the liquid field to the structure field.
Γ_{L1L5}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of liquid fuel to fuel particles.

Γ'_{L1L5}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of liquid fuel to fuel particles considering only the nonequilibrium model.
Γ_{L1S2}	Mass-transfer rate ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid fuel to frozen fuel.
Γ'_{L1S2}	Mass-transfer rate ($\text{kg}/\text{m}^3 \cdot \text{s}$) from liquid fuel to frozen fuel considering only the nonequilibrium model.
Γ_{L2F}	Total rate ($\text{kg}/\text{m}^3 \cdot \text{s}$) of freezing of liquid steel.
Γ_{L2L6}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of liquid steel to solid steel particles.
Γ_{L2S3}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of liquid steel to solid cladding.
Γ'_{L2S3}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of liquid steel to solid cladding considering only nonequilibrium effects.
Γ_{L2S4}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of liquid steel to solid can wall.
Γ'_{L2S4}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) of liquid steel to can wall considering only nonequilibrium effects.
Γ_m	Vaporization-condensation mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) for species m at an interface.
Γ_{mk}	Mass-transfer rate ($\text{kg}/\text{m}^3 \cdot \text{s}$) of material m on surface k .

Γ_{SkLm}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from structure energy-field component k to liquid energy-field component m.
Γ'_{SkLm}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from structure energy-field component k to liquid energy-field component m solely considering nonequilibrium surface effects.
Γ_{SL}	Total mass-transfer rate ($\text{kg}/\text{m}^3 \cdot \text{s}$) from the structure field to the liquid field.
Γ_{sm}	Total mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from structure-field density component m.
Γ_{SmF}	Total mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) because of breakup of structure-field energy component m.
Γ_{smgk}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from structure-field density component m to vapor-field density component k.
$\Gamma_{sm\ell k}$	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from structure-field density component m to liquid-field density component k.
Γ_{smsk}	Mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from structure-field density component m to structure-field density component k.
Γ_{GLm}^{k+1}	Total mass-transfer rate per unit volume ($\text{kg}/\text{m}^3 \cdot \text{s}$) from vapor-field energy component m to liquid-field energy component m after $k+1$ vaporization/condensation iterations.

$\underline{\Gamma_{Fm}^n}$	Vector of total mass-transfer rates per unit volume ($\text{kg/m}^3 \cdot \text{s}$) evaluated at the beginning of the current time step. The material is transferred from component m in either the liquid (l) or vapor (g) fields.
γ	Calculated value of the constant used in the quasi-static method to separate the neutron flux into amplitude and shape functions.
γ_c	Parameter used in the Chebyshev acceleration of the fission source solution.
γ_{cont}	Fraction of a cell's internal heat generation going into control material.
γ_{cont}	Fraction of the energy source deposited into Control components. (Used only for the Uranus option.)
γ_{Fert}	Fraction of the energy source deposited into Fertile components. (Used only for the Uranus option.)
γ_{Fiss}	Fraction of the energy source deposited into Fissile components. (Used only for the Uranus option.)
γ_{ijmg}	Denominator of the angular flux shape in the diamond-difference S_n method for neutronics mesh cell i, j ; angular interval m ; and energy group g (at time t).
$\gamma_{ij+1/2}$	Correction term added to the upward and downward neutron flows in the rebalance equation to improve stability of the scattering source iteration in the S_n transport solution.
$\gamma_{i+1/2j}$	Correction term added to the right-ward and left-ward neutron flows in the rebalance equation to improve stability of the scattering source iteration in the S_n transport solution.

γ_M	Ratio of the specific heats for vapor material M.
γ_m	Fraction of the energy source deposited into component m. (Used only for the Uranus option.)
γ_{Na}	Fraction of the energy source deposited into Sodium components. (Used only for the Uranus option.)
γ_{SS}	Fraction of the energy source deposited into Steel components. (Used only for the Uranus option.)
Δp	Pressure drop (Pa) over a distance given by L.
Δt	Fluid dynamics time step(s).
Δt	Neutronics time step(s). May refer to any of the shape time steps, reactivity time steps, or amplitude time steps.
$\tilde{\Delta t}$	Estimated fluid dynamics time step size(s) used in calculating exchange functions.
Δ_i and Δ_j	Shorthand notations for spatial differencing. Defined by Eqs. (IV-127) and (IV-128).
$\Delta_{i-1/2}$, Δ_{i+1} $\Delta_{j-1/2}$, Δ_{j+1}	Additional difference operators defined analogously to Δ_i , Δ_λ , Δ_j , and Δ_θ .
Δp_{max}	Maximum pressure change (Pa) used in the DTL(11) time-step control.
Δp_{ursf}	Pressure difference (Pa) across unrestructured fuel.
Δr_i	Width (m) of the i^{th} node in cylindrical or planar geometry.

Δr_i	Width (m) of the i^{th} radial column of neutronics mesh cells.
Δr_{i+1}	The mesh-cell width (m) of the $i+1^{\text{th}}$ node.
$\Delta r_{i+1/2}$	The mesh-cell width (m) averaged between Δr_i and Δr_{i+1} . Equivalent to Δr_λ .
$\Delta r_{i-1/2}$	The mesh-cell width (m) averaged between Δr_i and Δr_{i-1} .
Δr_{ursf}	Radial thickness (m) of the unrestructured fuel.
ΔT_G	Change in the vapor mixture temperature (K) if direct structure to vapor heat transfer is occurring.
Δt_{ri}	Trial time-step size(s) in the radial directions based on the Courant condition.
Δt_{zj}	Trial time-step size(s) in the axial direction based on the Courant condition.
Δv_M	Specific volume change (m^3/kg) during vaporization for material M.
Δx_i	Width (m) of the i^{th} node in planar geometry.
Δz_j	Height (m) of the j^{th} axial row of neutronics mesh cells.
Δz_j	Mesh-cell axial height (m) for the j^{th} node.
Δz_{j+1}	Mesh-cell axial height (m) for the $j+1^{\text{th}}$ node.
$\Delta z_{j+1/2}$	The mesh-cell length (m) averaged between Δz_j and Δz_{j+1} . Equivalent to Δz_θ .

$\Delta z_{j-1/2}$	The mesh-cell length (m) averaged between Δz_j and Δz_{j-1} .
$\Delta \underline{F}^k$	The change in $\underline{F}$ on iteration k . Evaluated by Eqs. (G-21) and (G-22). A j subscript is used to indicate one line of SIMMER-II elements in the overall vector that is of length $(\bar{I} \cdot \bar{J})$.
$\underline{\Delta Q}^k$	Vector giving the residual error in obtaining Q resulting from the $\underline{F}^k$ approximation to $\underline{F}$. Defined by Eq. (G-8). An element of this vector is subscripted by l .
Δt^h	Time interval from the start of the most recent reactivity step to the end of the most recent fluid-dynamics step.
Δt^n	Time-step size(s) for cycle n .
Δt^{n+1}	Advanced-time-step size(s), which can be based on any of a number of criteria.
Δt^{n-1}	Time-step size(s) for cycle $n-1$.
Δt^r	Time between reactivity time steps at t^r and t^{r-1} .
Δt^{r2}	Reactivity time step predicted on the basis of limiting the extrapolated reactivity change.
Δt^{r3}	Reactivity time step predicted on the basis of limiting the change in extrapolated flux amplitude.
Δt^s	Time between flux-shape time steps at t^s and t^{s-1} .
Δt^{s2}	Flux-shape time-step predicted on the basis of change in the leakage contribution to the reactivity.

Δt^{s8}	Flux-shape time step predicted on the basis of change in the spatial flux tilt.
$\underline{\Delta Q}_{est}^k$	Estimated change in Q from iteration k . Defined after Eq. (G-12). An element of this vector is subscripted with l .
$\underline{\Delta Q}_j^k$	A subvector containing the elements of $\underline{\Delta Q}^k$ for mesh cell line j .
$\underline{\Delta Q}_j^{+k}$	Modified $\underline{\Delta Q}_j^k$ vector. Defined by Eq. (G-16) for $j = 1$ and Eq. (G-20) for $j > 1$.
$\underline{\Delta Q}_j^{*k}$	The $\underline{\Delta Q}_j^{+k}$ vector premultiplied by $(\underline{P}_j^*)^{-1}$, as defined by Eq. (G-17).
Δt_t^{n+1}	Advanced-time-step size(s) based on the Courant conditions.
$\Delta \alpha_G$	Change in the vapor volume fraction used in the DTL(4) time-step control evaluated as $\alpha_G^n - \alpha_G^{n+1}$.
$\Delta \bar{p}_{Gm}$	Change in vapor density (kg/m^3) for vapor material m by the vaporization-condensation process between iterations κ and $\kappa+1$.
Δx	Symbolic minimum mesh cell dimension (m).
Δ_λ and Δ_θ	Shorthand notation for spatial differencing in momentum cells. Defined after Eq. (IV-172).
δ	Internally computed convergence parameter (kg/m^3) used in the vaporization-condensation model.

δ_{ijmg}	Correction term to the denominator of the angular flux shape for the weighted-diamond S_n method for neutronics mesh cell i, j ; angular interval m ; and energy group g (at time t).
δ_0	Input convergence parameter, EPHASE, (kg/m^3) used in the vaporization-condensation model.
$\delta\tilde{p}$	Change in pressure (Pa) during one pressure iteration. Can be subscripted to indicate i and j cell indices. Can be underlined to indicate a vector composed of all $(\tilde{i} \cdot \tilde{j})$ cells. A superscript κ can be used to indicate an iteration number.
$\delta\rho_{\text{Cont}}$	Control component displacement reactivity feedback used in the Uranus option.
$\delta\rho_{\text{Dop}}$	Doppler or fuel temperature reactivity feedback used in the Uranus option.
$\delta\rho_{\text{Fert}}$	Fertile fuel displacement reactivity feedback used in the Uranus option.
$\delta\rho_{\text{Fiss}}$	Fissile fuel displacement reactivity feedback used in the Uranus option.
$\delta\tilde{\rho}_L$	Small change in the total liquid macroscopic density (kg/m^3) .
$\delta\rho_m$	Change in microscopic density (kg/m^3) from the state at the single-phase/two-phase transition to a higher density single-phase state for component m .
$\delta\rho_{\text{NA}}$	Sodium displacement reactivity feedback used in the Uranus option.

$\delta\rho_p$	Driving or programmed reactivity.
$\delta\rho_p$	Programmed reactivity used in the Uranus option.
$\delta\rho_{SS}$	Steel displacement reactivity feedback used in the Uranus option.
$\delta\rho_{st}$	Stainless-steel displacement reactivity feedback.
δt^k	Small time step(s) used to integrate the equations of the multicomponent vaporization-condensation model.
$\delta V_{GL}^k $	Change in the velocity difference between the liquid and vapor fields during the k iteration.
$(\frac{\partial p}{\partial \rho_L})_{k \neq m}$	The inverse compressibility for the liquids in a cell when energy component m is not included.
ϵ	Convergence parameter for the SLOR and AIM matrix inversion procedures described in Appendix G.
ϵ	Estimated error in the eigenvalue for the current outer iteration of the neutron conservation equation solution.
ϵ	Estimated error in the integrated fission source for the current outer iteration of neutron conservation equation solution.
ϵ_{G1}	Input coefficient for pressure iteration convergence as used in Eq. (IV-218).
ϵ_{G2}	Input parameter ($\text{kg/m}^3 \cdot \text{s}$) for pressure iteration convergence as used in Eq. (IV-218).

ϵ_{L1}	Input coefficient for pressure iteration convergence as used in Eq. (IV-217).
ϵ_{L2}	Input parameter ($\text{kg/m}^3 \cdot \text{s}$) for pressure iteration convergence as used in Eq. (IV-217).
ϵ_L	Parameter used in the Chebyshev acceleration of the outer iteration solution of the neutron conservation equation.
ϵ_m	Input molecular force constant (K) for material m.
ϵ_0	Permitted error in the neutronics preprocessor (MXS) for fitting error of cross-section shielding functions as a function of background.
ϵ_1	Permitted error in the neutronics preprocessor (MXS) for fitting error of cross-section shielding functions as a function of temperature.
ϵ_2	Permitted error in the neutronic preprocessor (MXS) for fitting error for the temperature derivative of the cross-section shielding functions.
ϵ_λ	Error measure of the degree of convergence of the spatial fission source.
ξ	Greek symbol corresponding to I_{GEOM} . ξ is zero for plane geometry, whereas ξ is one for cylindrical geometry.
ξ_N	Normalization factor in the can-wall set-over model. Defined after Eq. (IV-84).

η	Parameter defining the mix of full and linearly interpolated donor-cell differencing to be used in a given SIMMER-II advection calculation. Subscripts indicate the real cell or momentum cell boundary that is being considered.
η_m	Discrete direction cosines of the neutron flux in the axial direction. The index refers to the angular interval m used in angular quadratures.
θ	Subscript representing the axial boundary of a momentum cell, $j+1/2$.
θ_m	Dimensionless temperature used in computing the collision integral for thermal conductivity for material m .
θ_{mk}	Dimensionless temperature used in computing the collision integral for diffusion involving materials m and k .
K	Unrestructured fuel permeability (m^2).
κ	Iteration index. Used as a superscript. Also used to count the number of small time steps required to integrate the equations of the multicomponent vaporization-condensation model.
k	Inner iteration index for the iteration between the mass continuity equation and the equation-of-state relationship between saturation temperature and vapor density.
Λ	Effective neutron generation time(s).
λ	User-specified exponent controlling drag as the liquid-field volume fraction increases.

λ	Subscript representing the radial (horizontal) boundary of a momentum cell, $i+1/2$.
$\underline{\lambda}$	Smallest ratio of two successive fission source iterates for all neutronics mesh cells.
$\bar{\lambda}$	Largest ratio of two successive fission source iterates for all neutronics mesh cells.
Λ_m	Fraction of structure surface area contacted by liquid energy component m.
λ_d	Decay constant (s^{-1}) for delayed neutron precursor group d.
λ_d	Time constant (s^{-1}) for decay of delayed neutron precursors in family d.
λ_{hi}	Decay constant (s^{-1}) for the i^{th} decay heat component.
Λ^r	Effective neutron generation time for the most recent reactivity step.
μ	Spectral radius of the iteration matrix used in the solution of the rebalance equation or the neutron diffusion equation.
μ	The iteration matrix spectral radius. Defined after Eq. (G-7).
μ_{fg}	Fission gas viscosity (Pa·s).
μ_G	Average viscosity (Pa·s) of the vapor mixture.
μ_{Gm}	Viscosity of vapor-field material component m.

μ_L	Average viscosity (Pa·s) input for the liquid field.
μ_{Lm}	Viscosity (Pa·s) of liquid-field energy component m.
μ_m	Discrete direction cosines of the neutron flux velocity vector in the radial direction. The index refers to the angular interval m used in angular quadratures.
μ_{mix}	Vapor mixture average viscosity (Pa·s).
μ_p	Particle viscosity (Pa·s) as input for the particulate components in the liquid field.
μ_ω	Estimated spectral radius of the iteration matrix used for neutronics solutions.
μ_1 and μ_2	Functional differences from above and below the approximated derivative.
ν_d	Delayed neutron yield for precursor group d.
ν_G	Vapor kinematic viscosity (m ² /s).
ν_L	Liquid kinematic viscosity (m ² /s).
$(\nu_p \Sigma_f)_{ijg}$	Product of the prompt neutron fission yield and the macroscopic fission cross section averaged over neutronics mesh cell ij and energy group g (at time t).
ν_g^i	Prompt neutron yield per fission for isotope i in energy group g.
ν_g^M	Prompt neutron yield per fission for cross-section material M in energy group g.

$\xi_{i-1,j}, \xi_{i+1,j}$ $\xi_{i,j-1}, \xi_{i,j+1}$	Indicator for a single-phase cell adjoining a two-phase cell used in Eq. (IV-135).
ξ_M	Exponent parameter for material M (input) in the heat-of-vaporization equation.
ξ_m	Parameter for liquid energy component m equal to one minus the reduced temperature. Used in Eq. (III-71).
π	Circumference of a circle divided by the circle's diameter.
$\bar{\pi}_G \cdot V_G$	Viscous component (Pa) in the vapor-momentum equation.
$\bar{\pi}_L \cdot V_L$	Viscous component (Pa) in the liquid-momentum equation.
ρ	Single-phase microscopic density (kg/m^3).
$\bar{\rho}$	Macroscopic density (kg/m^3) used to symbolically indicate participation in any of several processes, including the thermal response time of frozen material, calculation of liquid compressibility, and donor cell differencing in momentum cells. May be subscripted by i and j indices to indicate a cell location.
$\rho = \rho(t)$	Net reactivity.
$\bar{\rho}_{\text{Cont}}$	Total macroscopic density of control components in neutronics mesh cell i,j at time t.
$\bar{\rho}_{\text{cont}}$	In a control subassembly with a negative region parameter, the total macroscopic density (kg/m^3) of all materials to be modeled as control material.
$(\bar{\rho}_{\text{Cont}})_{ij}$	Total macroscopic density of control components in fluid-dynamics mesh cell i,j at time t. (Used only for the Uranus option.)

$\rho_{\text{Crt,M}}$	Critical density (kg/m^3).
ρ_{FD}	Delayed fission component of the net reactivity.
$\bar{\rho}_{\text{Fert}}$	Total macroscopic density of fertile components in neutronics mesh cell i,j at time t .
$(\bar{\rho}_{\text{Fert}})_{ij}$	Total macroscopic density of fertile components in fluid-dynamics mesh cell i,j at time t . (Used only for the Uranus option.)
$\rho_{\text{fg,sat}}$	Saturated value of fission gas on grain boundaries input as atoms/ m^2 . Stored as (kg/m^3).
$\bar{\rho}_{\text{Fiss}}$	Total macroscopic density of fissile components in neutronics mesh cell i,j at time t .
$(\bar{\rho}_{\text{Fiss}})_{ij}$	Total macroscopic density of fissile components in fluid-dynamics mesh cell i,j at time t . (Used only for the Uranus option.)
ρ_{FP}	Prompt fission component of the net reactivity.
ρ_{G}	Vapor microscopic density (kg/m^3).
$\bar{\rho}_{\text{G}}$	Vapor-field macroscopic density (kg/m^3). κ superscripts can be used to indicate the iteration number. Subscripts i and j can be used to indicate a cell location, whereas λ and θ subscripts indicate a cell average.
$\hat{\rho}_{\text{G}}$	Estimated vapor mixture macroscopic density (kg/m^3) for time $\tilde{t}^{n+1}$ computed without accounting for convection.

$\bar{\rho}_G$	Total intermediate macroscopic density (kg/m^3) for the vapor field. Can be subscripted by i and j to indicate a cell location and superscripted by a Greek κ to indicate the iteration number in the pressure iteration.
$(\bar{\rho}_G e_G)^+$	Updated vapor energy per unit volume (J/m^3) without the addition of convective terms. When underlined, a vector of elements $\bar{I} \cdot \bar{J}$ in length is represented.
$(\bar{\rho}_G e_G)^{n+1}$	Final advanced-time vapor energy per unit volume (J/m^3). When underlined, a vector of elements $\bar{I} \cdot \bar{J}$ in length is represented.
ρ_{Gm}	Microscopic density (kg/m^3) of vapor material m . κ or k superscripts can be used to indicate the iteration number.
$\bar{\rho}_{Gm}$	Macroscopic density (kg/m^3) of vapor material m . k superscripts can be used to indicate the iteration number.
$(\bar{\rho}_G u_G)_\lambda$	Explicit vapor-field momentum contribution at the right cell boundary ($i+1/2, j$).
$(\bar{\rho}_G v_G)_\theta$	Explicit vapor-field momentum contribution at the top cell boundary ($i, j+1/2$).
$(\rho_{Gm} T_G)_{\text{Crt}}$	Product of the critical density and critical temperature ($\text{kg} \cdot \text{K/m}^3$) for material m .
$(\rho_{Gm} T_G)_0$	Product of the vapor density and temperature ($\text{kg} \cdot \text{K/m}^3$) for material m at a saturation temperature that is 95% of the critical temperature.

ρ_{gm}	Microscopic density (kg/m^3) of vapor-field density component m.
$\bar{\rho}_{gm}$	Macroscopic density (kg/m^3) of vapor-field density component m. The cell location can be indicated by i and j indices.
$\tilde{\rho}_{gm}$	Intermediate macroscopic density (kg/m^3) for vapor-field density component m.
ρ_L	Leakage component of the net reactivity.
ρ_L	Total liquid-field microscopic density (kg/m^3). For a single-component system, this is simply the liquid density reported in the literature.
$\bar{\rho}_L$	Total liquid-field macroscopic density. Subscripts i and j can be used to indicate a cell location, whereas λ and θ can be used to indicate a cell average.
$\tilde{\rho}_L$	Intermediate value (following the explicit update) of the macroscopic density (kg/m^3) summed over the liquid field. Can be subscripted by i and j to indicate a cell location and superscripted by a κ to indicate a iteration number in the pressure iteration.
$\rho_{L,\text{eff}}$	Effective liquid field microscopic density (kg/m^3). For a single-component system, the deviation of $\rho_{L,\text{eff}}$ from ρ_L determines the pressure increase above saturation values.
$\rho_{l\ell m}$	Microscopic density (kg/m^3) of the pure liquid phase of liquid-field density component m.
ρ_{Lm}	Microscopic density (kg/m^3) of liquid-field density component m.

$\bar{\rho}_{Lm}$	Macroscopic density (kg/m^3) for liquid-field energy component m.
$\tilde{\rho}_{Lm}$	Intermediate value (following the explicit update) of the macroscopic density (kg/m^3) of liquid-field energy component m.
ρ_{Lm}	Microscopic density (kg/m^3) of liquid-field density component m.
$\bar{\rho}_{Lm}$	Macroscopic density (kg/m^3) of liquid-field density component m. The cell location can be indicated by i and j indices.
$\tilde{\rho}_{Lm}$	Intermediate value (following the explicit update) of the macroscopic density (kg/m^3) of liquid-field density component m.
$\bar{\rho}_{Lo}$	Macroscopic density (kg/m^3) for the liquid field at the single-phase/two-phase transition.
ρ_{Lsm}	Microscopic density (kg/m^3) of pure solid phase of liquid-field density component m.
$(\bar{\rho}_L^U)_\lambda$	Explicit liquid-field momentum contribution at the right cell boundary ($i+1/2, j$).
$(\bar{\rho}_L^V)_\theta$	Explicit liquid-field momentum contribution at the top cell boundary ($i, j+1/2$).
$\bar{\rho}_m$	Macroscopic density (kg/m^3) for energy component m.
ρ_{ma}	Density (kg/m^3) of material m below an interface.
ρ_{mb}	Density (kg/m^3) of material m above an interface.

$\bar{\rho}_{Na}$	Total macroscopic density of sodium components in neutronics mesh cell i,j at time t.
$(\bar{\rho}_{Na})_{ij}$	Total macroscopic density of sodium components in fluid-dynamics mesh cell i,j at time t. (Used only for the Uranus option.)
ρ_S	Scatter component of the net reactivity.
ρ_{slm}	Microscopic density (kg/m^3) of the pure liquid phase of structure-field density component m.
$\bar{\rho}_{Sm}$	Macroscopic density (kg/m^3) for structure-field energy component m. Additional i and j subscripts can be used to identify a cell location.
ρ_{sm}	Microscopic density (kg/m^3) of structure-field density component m.
$\bar{\rho}_{sm}$	Macroscopic density (kg/m^3) of structure-field density component m. The cell location can be indicated by i and j indices.
$\bar{\rho}_{SS}$	Total macroscopic density of steel components in neutronics mesh cell i,j at time t.
$(\bar{\rho}_{SS})_{ij}$	Total macroscopic density of steel components in fluid-dynamics mesh cell i,j at time t.
ρ_{ssm}	Microscopic density (kg/m^3) of pure solid phase of structure-field density component m.
$(\bar{\rho}_{s5})_{fail}$	Minimum cladding macroscopic density (kg/m^3) above which cladding is assumed to be intact.

$\bar{\rho}_{S6,i\pm 1,j}$	The sum of the can-wall macroscopic densities (kg/m^3) in cells (i+1,j) and (i-1,j).
ρ_{s9}	Intergranular fission gas density (kg/m^3) in the unstructured fuel.
ρ_T	Total (associated with total cross section) component of the net reactivity.
ρ^r	Net reactivity at the most recent reactivity step.
ρ^s	Net reactivity at the most recent shape step.
$\bar{\rho}_{\text{cont}}^{\text{cont}}$	In a control subassembly with a negative region parameter, the macroscopic density (kg/m^3) of control material.
$\bar{\rho}_L^i$	Total liquid-field macroscopic density as input by the user.
$\bar{\rho}_{Lm}^i$	Macroscopic density (kg/m^3) for liquid-field energy component m as input by the user.
$\bar{\rho}_{\ell m}^i$	Macroscopic density (kg/m^3) for liquid-field density component m as input by the user.
$\bar{\rho}_{ij}^m$	Average density of component m for neutronics mesh cell i,j.
$\bar{\rho}_{Fm}^n$	Vector of macroscopic densities (kg/m^3) at time step n, for density component m and for either the liquid (ℓ) or vapor (g) field.
$\bar{\rho}_{Fm}^{n+1}$	Vector of macroscopic densities (kg/m^3) at time step n+1, for density component m and for either the liquid (ℓ) or vapor (g) field.

$\bar{\rho}_G^n$	Current (beginning of time step) vapor-field macroscopic density (kg/m^3). Both i and j subscripts may be used to denote the cell identification.
ρ_G^{n+1}	Advanced time total microscopic density (kg/m^3) for the vapor field.
ρ_{Gm}^n	Current (beginning of time step) microscopic vapor density (kg/m^3) of vapor material m.
$\bar{\rho}_{Gm}^n$	Current (beginning of time step) macroscopic density (kg/m^3) of vapor material m.
ρ_{Gm}^{n+1}	Advanced time microscopic vapor density (kg/m^3) of vapor material m.
$\bar{\rho}_{Gm}^{n+1}$	Advanced time macroscopic density (kg/m^3) for vapor material m.
$\bar{\rho}_{gm}^n$	Current (beginning of time step) macroscopic density (kg/m^3) for vapor-field density component m.
$\bar{\rho}_{gm}^{n+1}$	Advanced-time macroscopic density (kg/m^3) for vapor-field density component m.
$\bar{\rho}_L^n$	Current (beginning of time step) value for the macroscopic liquid density summed over the liquid field. Both i and j subscripts may be used to denote the cell identification.
$\tilde{\rho}_L^n$	Intermediate value for the total liquid-field macroscopic density (kg/m^3) obtained by ignoring convection.
$\bar{\rho}_L^{n+1}$	Advanced-time value for the total liquid-field macroscopic density (kg/m^3).

$\bar{\rho}_{Lm}^n$	Current macroscopic density (kg/m^3) of liquid-field energy component m.
$\tilde{\rho}_{Lm}^n$	Current macroscopic density (kg/m^3) of liquid-field energy component m after application of the non-equilibrium freezing and melting rates and vaporization. Used in Sec. III.
$\tilde{\bar{\rho}}_{Lm}^n$	Intermediate macroscopic density (kg/m^3) of liquid-field energy component m obtained by ignoring convection. Used in Sec. IV.
$\bar{\rho}_{\lambda m}^n$	Current (beginning of time step) value of macroscopic density (kg/m^3) for liquid density component m.
$\rho_{\lambda m}^{n+1}$	Advanced-time microscopic densities (kg/m^3) of liquid-field density component m.
$\bar{\rho}_{Lm}^{n+1}$	Advanced-time value of macroscopic density (kg/m^3) of liquid-field energy component m.
$\bar{\rho}_{\text{cont}}^{\text{Na}}$	In a control subassembly with a negative region parameter, the macroscopic density (kg/m^3) of sodium modeled as control material.
$\bar{\rho}_{L, \text{est}}^{n+2}$	Estimated liquid-field macroscopic density at the end of time step n+2 as computed by Eq. (IV-206).
$\tilde{\bar{\rho}}_{Sk}^n$	Current macroscopic density (kg/m^3) of structure-field energy component k after application of the nonequilibrium freezing and structure breakup rates.
$\bar{\rho}_{Sm}^n$	Current macroscopic density (kg/m^3) of structure-field energy component m.

$\bar{\rho}_{sm}^{n+1}$	Advanced time macroscopic density (kg/m^3) for structure-field energy component m.
$\bar{\rho}_{sm}^n$	Current (beginning of time step) macroscopic density (kg/m^3) of structure-field density component m.
ρ_{sm}^{n+1}	Advanced-time microscopic density (kg/m^3) of structure-field density component m.
$\bar{\rho}_{sm}^{n+1}$	Advanced-time macroscopic density (kg/m^3) of structure-field density component m.
$(\bar{\rho}_{\text{Cont}}^0)_{ij}$	Total macroscopic density of control components in fluid-dynamics mesh cell i,j (at time $t = 0$). (Used only for the Uranus option.)
$(\bar{\rho}_{\text{Fert}}^0)_{ij}$	Total macroscopic density of fertile components in fluid-dynamics mesh cell i,j (at time $t = 0$). (Used only for the Uranus option.)
$(\bar{\rho}_{\text{Fiss}}^0)_{ij}$	Total macroscopic density of fissile components in fluid-dynamics mesh cell i,j (at time $t = 0$). (Used only for the Uranus option.)
$(\bar{\rho}_{\text{Na}}^0)_{ij}$	Total macroscopic density of sodium components in fluid-dynamics mesh cell i,j (at time $t = 0$). (Used only for the Uranus option.)
$(\bar{\rho}_{\text{SS}}^0)_{ij}$	Total macroscopic density of steel components in fluid-dynamics mesh cell i,j (at time $t = 0$). (Used only for the Uranus option.)
$\bar{\rho}_{\text{Gm}}^0$	Macroscopic density (kg/m^3) of vapor material m as given by the input processor.

$\bar{\rho}_{Lm}^0$	Macroscopic density (kg/m^3) for liquid-field energy component m as determined by the input processor.
$\bar{\rho}_{\lambda m}^0$	Macroscopic density (kg/m^3) for liquid-field density component m as adjusted by the input processor.
ρ_{s8}^0	Initial intragranular fission gas density (kg/m^3) in unrestructured fuel.
$\bar{\rho}_{\text{cont}}^{\text{steel}}$	In a control subassembly with a negative region parameter, the macroscopic density (kg/m^3) of steel modeled as control material.
$\Sigma_{r,ijg}$	Macroscopic removal cross section (m^{-1}) averaged over neutronics mesh cell i,j and energy group g (at time t).
$\Sigma_{s,ijg \rightarrow g}$	Macroscopic scattering cross section (m^{-1}) averaged over neutronics mesh cell i,j and group g for neutron transfer from group g' to group g at time t.
$\Sigma_{t,ijg}$	Macroscopic total cross section (m^{-1}) averaged over neutronics mesh cell i,j and energy group g at time t.
$\Sigma_{x,ijg}$	Macroscopic cross section (m^{-1}) for reaction rate type x for neutronics mesh cell i,j in energy group g at time t.
$\Sigma_f(t,r,z,E)$	Macroscopic fission cross section as a function of time, cylindrical coordinates, and energy (m^{-1}).
$\Sigma_s(t,r,z,E \rightarrow E')$	Macroscopic cross section (m^{-1}) for isotropic neutron scattering from neutron energy E' to neutron energy E as a function of time, cylindrical coordinates, and energy.

$\Sigma_t = \Sigma_t(t, r, z, E)$	Macroscopic total cross section as a function of time, cylindrical coordinates, and energy (m^{-1}).
$\Sigma_c^M(t, r, z, E')$	Macroscopic capture cross section of material M as a function of time, cylindrical coordinates, and energy (m^{-1}).
$\Sigma_f^M(t, r, z, E')$	Macroscopic fission cross section of material M as a function of time, cylindrical coordinates, and energy (m^{-1}).
Σ_e^m, ijg	Macroscopic energy deposition rate per unit density (W/kg) of component m for neutronics mesh cell i, j in group g at time t.
$\bar{\sigma}$	Estimate of the eigenvalue dominance ratio used in the outer iteration solution for the neutron conservation equations.
σ_{Lm}	Interfacial surface tension (kg/s^2) between the vapor and liquid-field energy component m.
σ_m	Characteristic diameter ($\overset{0}{\text{\AA}}$) of the molecule.
σ_{mk}	Characteristic diameter ($\overset{0}{\text{\AA}}$) used in the relative diffusion of materials m and k.
σ_0	Eigenvalue dominance ratio used in the outer iteration solution of the neutron conservation equations.
$\sigma_y(T_s)$	Yield stress (Pa) for can-wall material at temperature T_s .
σ^{cont}	Effective microscopic cross section (input as barns) for control material in a control subassembly input with a negative region parameter.

σ^{cont}	Microscopic control cross section (barns) in a control subassembly composition.
σ^{Na}	Microscopic sodium cross section (barns) in a control subassembly composition.
σ^{steel}	Microscopic steel cross section (barns) in a control subassembly composition.
$\sigma_{\text{c},ijg}^i$	Shielded microscopic capture cross section for isotope i for mesh cell i,j in energy group g.
$\sigma_{\text{f},ijg}^i$	Shielded microscopic fission cross section for isotope i for mesh cell i,j in energy group g.
$\sigma_{\text{t},g}^i$	Infinitely dilute (unshielded) total microscopic cross section for isotope i in energy group g.
$\sigma_{\text{t},ijg}^i$	Shielded microscopic total cross section for isotope i for neutronics mesh cell i,j in energy group g.
$\sigma_{\text{x},g}^i$	Infinitely dilute (unshielded) microscopic cross section for reaction type x for isotope i in energy group g.
$\sigma_{\text{x},ijg}^i$	Shielded microscopic cross section for reaction type x for isotope i for neutronic mesh cell i,j in energy group g.
$\sigma_{\text{o},ijg}^j$	Background cross section per atom of isotope j in neutronics mesh cell i,j in energy group g.
$\sigma_{\text{c},g}^k$	Infinitely dilute (unshielded) microscopic capture cross section for isotope k in group g.

$\sigma_{e,g \rightarrow g}^k$	Infinitely dilute (unshielded) microscopic elastic scatter cross section for isotope k from group g to group g.
$\sigma_{e,g \rightarrow g+1}^k$	Infinitely dilute (unshielded) microscopic elastic downscatter cross section for isotope k from group g to group g+1.
$\sigma_{f,g}^k$	Infinitely dilute (unshielded) microscopic fission cross section for isotope k in group g.
$\sigma_{msc,g}^k$	Microscopic cross section for the combination of (n, α), (n,p), (n,D), and (n,T) reactions for isotope k in group g.
$\sigma_{ne,g \rightarrow g}^k$	Microscopic cross section for the combined in-group inelastic scatter and in-group (n,2n) scatter for isotope k in group g.
$\sigma_{r,g}^k$	Microscopic removal cross section for the combination of the total inelastic downscatter, the in-group (n,2n) scatter, and the total (n,2n) downscatter for isotope k in group g.
$\sigma_{x,g}^M$	Infinitely dilute (unshielded) microscopic cross section for reaction type x for material M in group g.
$\sigma_{x,ijg}^M$	Shielded microscopic cross section for reaction type x for material M for neutronics mesh cell i,j in energy group g.
$\sigma_{o,ijg}^{n+1,j}$	n+1 iterate of the background cross section per atom of isotope j for mesh cell i,j in group g.
τ	Thermal response time (s) of frozen material or of a small liquid droplet.

$\Phi(t,r,z,E)$	Neutron scalar flux as a function of time, cylindrical coordinates, and energy. The neutron flux is given by the product of the flux shape and the amplitude function.
$\Phi^*(t,r,z,E)$	Adjoint neutron scalar flux as a function of time, cylindrical coordinates, and energy.
$\phi(t,r,z,E)$	Neutron scalar flux shape as a function of time, cylindrical coordinates, and energy. The neutron flux is given by the product of the flux shape and the amplitude function.
Φ_{ijg}^*	Adjoint neutron scalar flux averaged over neutronics mesh cell ij and integrated over energy group g (at time $t = 0$).
ϕ_{ijg}	Neutron flux shape averaged over neutronics mesh cell i,j and integrated over energy group g at time t . (May refer to the flux shape as obtained by solution of the diffusion equation or by integration of the angular flux shape from the S_n transport equation.)
ϕ_{mk}	Dimensionless functions used to compute the thermal conduction and viscosity of gas mixtures.
ϕ_{ijg}^{k-1}	Neutron flux shape averaged over neutronics mesh cell i,j and integrated over energy group g at time step $k-1$.
$x_d(E)$	Delayed neutron emission spectrum from delayed neutron precursor family d as a function of energy.
x_{dg}	Delayed neutron emission fraction from delayed neutron precursor family d into energy group g .

χ_{pg}	Prompt neutron fission emission spectrum for the isotope or material to be used to represent all isotopes or materials in group g.
χ_{pg}^i	Prompt neutron fission emission spectrum for isotope i into group g.
χ_{pg}^M	Prompt neutron emission spectrum for material M into group g.
$\Psi = \Psi(t, r, z, \vec{\Omega}, E)$	Neutron flux as a function of time, cylindrical coordinates, direction, and energy.
$\Psi^* = \Psi^*(t, r, z, \vec{\Omega}, E)$	Adjoint angular flux as a function of time, cylindrical coordinates, direction, and energy.
$\psi = \psi(t, r, z, \vec{\Omega}, E)$	Neutron flux shape as a function of time, cylindrical coordinates, direction, and energy.
Ψ_{ijmg}	Neutron flux averaged over neutronics mesh cell i,j, representing angular interval m, and integrated over energy group g at time t.
Ψ_{ijmg}^*	Adjoint angular flux averaged over neutronics mesh cell i,j representing angular interval m and integrated over energy group g (at time t = 0).
ψ_{ijmg}	Neutron flux shape averaged over neutronics mesh cell i,j representing angular interval m, and integrated over energy group g at time t.
ψ_p	Internal multiplier on the DTL(8) time-step control for condensation. Set to 1 or 2 depending on whether cell outflow is anticipated to result in a negative density.

ω	Overrelaxation parameter in the SLOR and AIM matrix inversion algorithm discussed in Appendix G.
ω	Overrelaxation parameter used in the matrix solution algorithm for solution of the rebalance equation or the neutron diffusion equation.
ω	Proportionality constant between the rate of change of the droplet number density and the collision frequency.
ω_{previous}	The value of the relaxation parameter based on its last update.
ω^r	Inverse period at the most recent reactivity time step.
ω^s	Inverse period at the most recent shape time step.
$\Omega_{D,mk}$	Molecular collision integral for the relative diffusion of materials m and k.
$\Omega_{k,m}$	Molecular collision integral for thermal conductivity for material m.
Ω_{ij}^{ℓ}	The ℓ^{th} iterate of the fission source solution for neutronics mesh cell i,j at time t.
$\Omega_{u,ij}^{\ell}$	The ℓ^{th} iterate of the fission source solution (unnormalized) for neutronics mesh cell i,j at time t.
$\tilde{\Omega}_{ij}^{\ell+1}$	The fission source guess for the $\ell+1^{\text{th}}$ iteration of the neutron conservation equation for neutronics mesh cell i,j at time t.
$\diamond$	Brackets used to denote donor cell differencing. These may be subscripted to indicate the spatial interface and superscripted to indicate the temporal step.

APPENDIX B

SINGLE-PHASE EQUATION OF STATE

In SIMMER-II, the transition from a single- to a two-phase flow calculation occurs when the vapor volume fraction exceeds a specified percentage of the available flow volume fraction. The available flow volume fraction is given by $1 - \alpha_S$, where α_S is the structure-field total volume fraction. The fraction of this that determines the transition between single- and two-phase flow is denoted by α_0 , that is,

$$\alpha_G > \alpha_0 (1 - \alpha_S) \quad \text{indicates a mesh cell having two-phase flow, and} \quad (\text{B-1})$$

$$\alpha_G < \alpha_0 (1 - \alpha_S) \quad \text{indicates a mesh cell having single-phase flow.} \quad (\text{B-2})$$

For mesh cells with two-phase flow, the cell pressure is determined from the vapor EOS, for example, the modified perfect-gas EOS, if the analytic EOS is used. If the vapor volume fraction cutoff, α_0 , is very small, the pressure calculation from the vapor EOS can be very sensitive to changes in the vapor volume fraction as the transition between single- and two-phase flow is approached. Thus, α_0 equal to zero is never used, and most often a value of 0.05 is input to the SIMMER-II code.

I. SINGLE-COMPONENT SYSTEMS

At the single-phase/two-phase transition it is necessary to have a continuous thermodynamic state. Otherwise, a mesh cell could oscillate between single- and two-phase flow. Because the pressure determines the fluid-dynamics state, pressure continuity must be maintained. In addition, thermal continuity is necessary so that large temperature gradients do not induce high phase-transition rates. Continuity in pressure and temperature implies that the liquid and vapor are in equilibrium at the transition from two-phase to single-phase flow. That is, for a single-component system,

$$P_L = P_V = P_G = \rho_G R_M T_G \quad , \quad (\text{B-3})$$

and

$$T_L = T_G \quad , \quad (B-4)$$

where

p_L is the single-phase liquid pressure,

p_v is the vapor pressure,

p_G is the vapor-state pressure determined from the EOS,

ρ_G is the vapor-field component microscopic density,

R_M is the gas constant for the material M under consideration,

T_G is the vapor-field component temperature, and

T_L is the liquid-field component temperature.

In SIMMER-II, the vapor pressure p_v is given by the two-parameter fit

$$p_v (T_{sat}) = p^* e^{-T^*/T_{sat}} \quad , \quad (B-5)$$

where p^* and T^* are input parameters determined from a fit to appropriate two-phase data and T_{sat} is the saturation temperature for the pressure p_v .

To develop the single-phase EOS, a single-component system is considered first, and the treatment then is extended to a multicomponent system. Equation (B-3) indicates the appropriate liquid pressure at the single-phase/two-phase transition, that is,

$$p_L = p_v (T_L) \quad \text{when } \alpha_G = \alpha_{Go} = \alpha_o (1 - \alpha_S) \quad . \quad (B-6)$$

To develop the EOS further, liquid must be compressed to add more liquid to the mesh cell, thus increasing the pressure in the cell as shown in Fig. B-1. In this pressure-volume diagram, point A indicates the pressure-density state at the single-phase/two-phase transition, and point B is the state achieved through compression of the liquid. That is,

Fig. B-1.
Pressure-volume relation for the single-phase EOS.

$$P_L = P_v(T_L) + \frac{\partial P_L}{\partial v_L} \delta v_L \quad , \quad \text{or} \quad (B-7)$$

$$P_L = P_v(T_L) + \frac{\partial P_L}{\partial \rho_L} \delta \rho_L \quad . \quad (B-8)$$

The microscopic density is used instead of the specific volume because SIMMER-II uses densities. The derivative $\frac{\partial P_L}{\partial \rho_L}$ is the square of the liquid-field sonic velocity, c_L , and is assumed to be constant in SIMMER-II except for the case of liquid fuel, where it depends on the liquid temperature. Thus, Eq. (B-8) becomes

$$p_L = p_v (T_L) + c_L^2 (\rho_{L,eff} - \rho_L) \quad , \quad (B-9)$$

where $\rho_{L,eff}$ is the effective liquid-field component microscopic density and ρ_L is the liquid-field component microscopic density at the single-phase/two-phase transition. Because the normal microscopic density used in SIMMER-II determines the single-phase/two-phase transition point, ρ_L is the normal microscopic density. Because the normal microscopic density generally is desired to be associated with a liquid's density reported in the literature, for example, as in steam tables, we define the liquid volume fraction, α_L , by

$$\frac{\bar{\rho}_L}{\rho_L} (1 - \alpha_0) = \alpha_L \quad , \quad (B-10)$$

where $\bar{\rho}_L$ is the liquid-field component macroscopic density. Consequently, at the single-phase/two-phase transition, $\bar{\rho}_L$ becomes $\bar{\rho}_{Lo}$, which is given by

$$\bar{\rho}_{Lo} = (1 - \alpha_S) \rho_L \quad , \quad (B-11)$$

and the liquid-field component macroscopic density is the same value as if it occupied all the volume available for fluid in a cell. This definition means that $\rho_{L,eff}$ is given by $\bar{\rho}_L/(1 - \alpha_S)$ or the mass per unit of available fluid volume. Consequently, we may write

$$p_L = p_v (T_L) + c_L^2 \left[\frac{\bar{\rho}_L}{1 - \alpha_S} - \rho_L \right] \quad . \quad (B-12)$$

Using the definition of the volume fraction at the single-phase/two-phase transition, α_{Lo} , we obtain

$$\alpha_{Lo} = 1 - \alpha_S - \alpha_{Go}$$

$$= 1 - \alpha_S - \alpha_o (1 - \alpha_S) \quad (B-13)$$

$$= (1 - \alpha_o) (1 - \alpha_S) \quad ,$$

which gives [by use of Eq. (B-10)]

$$p_L = p_v (T_L) + \frac{c_L^2}{1 - \alpha_S} \left[\frac{\alpha_L - \alpha_{Lo}}{\alpha_L} \right] \bar{p}_L \quad , \quad (B-14)$$

or

$$p_L = p_v (T_L) + \frac{c_L^2}{\alpha_L} \left[\frac{\alpha_o (1 - \alpha_S) - \alpha_G}{(1 - \alpha_S)} \right] \bar{p}_L \quad . \quad (B-15)$$

In addition, the SIMMER-II fluid-dynamics method needs the change in pressure with respect to macroscopic density. This is obtained from Eq. (B-12) as

$$\frac{\partial p}{\partial \bar{p}_L} = \frac{c_L^2}{1 - \alpha_S} \quad . \quad (B-16)$$

II. MULTICOMPONENT SYSTEMS

For a multicomponent system, the components are assumed to be immiscible. Therefore, the mesh-cell pressure at the transition is the sum of the liquid-field component pressures as given by the component vapor-pressure curves. Additional mass added to a mesh cell again causes compression of the liquid, increasing the pressure. For a given liquid-field component m , the pressure in that component is

$$p_{Lm} = \sum_k p_{vk} (T_{Lk}) + c_{Lm}^2 \delta \rho_m \quad , \quad (B-17)$$

where

p_{Lm} is the pressure in liquid-field component m,

c_{Lm} is the square of the sonic velocity for liquid-field component m,
and

$\delta\rho_m$ is the change in microscopic density from the state at the single-phase/two-phase transition to the current liquid state.

To obtain the single-phase pressure, all liquid-field component pressures are assumed to be equal, and the sum of the macroscopic densities is assumed to be the total macroscopic density for the liquid field. First,

$$\hat{p}_L = \sum_k p_{vk} (T_{Lk}) \quad (B-18)$$

is defined. Then, by assuming that the volume occupied by each component is proportional to its volume fraction, the multicomponent equivalent to Eq. (B-12) is

$$p = p_{Lm} = \hat{p}_L + c_{Lm}^2 \left[\frac{\bar{p}_{Lm}}{\frac{\alpha_{Lm}}{\alpha_L} (1 - \alpha_S)} - \rho_{Lm} \right] \quad (B-19)$$

where the m subscript has been added to indicate component dependence. Equation (B-19) is multiplied by α_{Lm}/c_{Lm}^2 and summed over m to yield

$$(p - \hat{p}_L) \sum_m \frac{\alpha_{Lm}}{c_{Lm}^2} = \sum_m \left(\frac{\alpha_L \bar{p}_{Lm}}{1 - \alpha_S} - \alpha_{Lm} \rho_{Lm} \right) \quad (B-20)$$

The multicomponent version of Eq. (B-10) is

$$\frac{\bar{p}_{Lm}}{\rho_{Lm}} (1 - \alpha_o) = \alpha_{Lm} \quad (B-21)$$

Substitution of Eq. (B-21) into Eq. (B-20) gives

$$p = \hat{p}_L + \frac{1}{\sum_m \frac{\alpha_{Lm}}{c_{Lm}^2}} \left[\frac{\alpha_L}{1 - \alpha_S} - (1 - \alpha_O) \right] \sum_m \bar{p}_{Lm} . \quad (B-22)$$

Use of the definition for α_G allows Eq. (B-22) to be written as

$$p = \hat{p}_L + \frac{1}{\sum_m \frac{\alpha_{Lm}}{c_{Lm}^2}} \left[\frac{\alpha_O (1 - \alpha_S) - \alpha_G}{1 - \alpha_S} \right] \bar{p}_L , \quad (B-23)$$

which is similar to Eq. (B-15). By using the definition of α_{Lo} [Eq. (B-13)] in Eq. (B-22), we obtain Eq. (III-58) in the text, where the square of the effective sonic velocity is given by

$$\frac{1}{\bar{c}^2} = \sum_m \frac{\alpha_{Lm}}{\alpha_L c_{Lm}^2} . \quad (B-24)$$

In the current version of SIMMER-II, the pressure iteration is performed assuming all single-phase cells retain constant composition. By assuming the α_{Lm}/α_L are constants, Eq. (B-20) can be differentiated to obtain

$$\frac{\partial p}{\partial \bar{p}_L} = \frac{\bar{c}^2}{1 - \alpha_S} . \quad (B-25)$$

This is evaluated in the code by using the definition of α_{Lm} [Eq. (B-21)], and of α_{Lo} [Eq. (B-13)] to yield

$$\frac{\partial p}{\partial \bar{\rho}_L} = \frac{\alpha_L}{\alpha_{Lo}} \frac{1}{\sum_m \frac{\bar{\rho}_{Lm}}{\rho_{Lm} c_{Lm}^2}} \quad . \quad (B-26)$$

Under some conditions, SIMMER-II will compute that a small amount of material at a high temperature is present in a mesh cell. Thus, the sum of the liquid pressures, $\bar{p}_L$ in Eq. (B-18), can be primarily a result of small amount of liquid. To offset this anomaly, the following arguments are made.

The high pressure of a small amount of material will compress the other material in a cell. If this pressurization were considered to come from additional single-phase mass added to the mesh cell (that is, not from the vapor pressure of the small amount of material), then the corresponding density increase can be found from

$$p_{vk}(T_{Lk}) = \left(\frac{\partial p}{\partial \bar{\rho}_L} \right)_{m \neq k} \delta \bar{\rho}_L \quad , \quad (B-27)$$

where

$$\left(\frac{\partial p}{\partial \bar{\rho}_L} \right)_{m \neq k} = \frac{1}{\sum_{m \neq k} \frac{\alpha_{Lm}}{\alpha_L c_{Lm}^2} (1 - \alpha_S)} \quad .$$

If the macroscopic density, $\bar{\rho}_{Lk}$, of the small amount of material is less than $\delta \bar{\rho}_L$ in Eq. (B-27), then the small amount of material must vaporize to occupy the volume, decreasing its contribution to the total vapor pressure. Consequently, for the analytic EOS, the pressure used in Eq. (B-18) for component k is the minimum of

$$p_{vk}(T_{Lk}) = p_M^* e^{-T_M^*/T_{Lk}} \quad , \quad (B-28)$$

the vapor-pressure relationship;

$$p_{vk}(T_{Lk}) = \bar{\rho}_{Lk} \left(\frac{\partial p}{\partial \bar{\rho}_L} \right)_{m \neq k} , \quad (B-29)$$

for replacement by compressed liquid; or

$$p_{vk}(T_{Lk}) = [(\bar{\rho}_{Gk} + \bar{\rho}_{Lk}) R_{M(k)} T_G] / \alpha_G , \quad (B-30)$$

for complete vaporization,

where

$\bar{\rho}_{Gk} + \bar{\rho}_{Lk}$ is the total macroscopic density of component k ,

$R_{M(k)}$ is the gas constant for component k ,

T_G is the vapor temperature, and

M is the material corresponding to liquid energy component k .

Finally, SIMMER-II treats the liquid sonic velocity for density components rather than energy components. The original intent of this was to allow different sonic velocities for fertile and fissile fuel. However, because the most recent version of SIMMER-II allows only one temperature-dependent function for the fuel sonic velocity, the density component treatment has become irrelevant, and Appendix B has not been complicated by these considerations. Equation (IV-30) illustrates how Eq. (B-22) could look when written in terms of density components and solved for an α_L based on an assumed pressure, p^0 .

APPENDIX C

COALESCENCE MODEL

The coalescence model is based on droplet-droplet random collisions that result in coalescence and, consequently, a reduction in the number density of droplets and an increase in the size of the remaining droplets. Figure C-1 illustrates the basis for the collision frequency. The liquid droplets of a particular material, m, are assumed to have an average random velocity component, $\bar{V}_m$, in a given region of space. This random component is assumed to be small relative to the liquid-field velocity, $\vec{V}_L$. In a time interval, Δt , a droplet of radius r_{pm} sweeps a volume, VOL_{s_m} , given by

$$VOL_{s_m} = \pi r_{pm}^2 (\bar{V}_m \Delta t) \quad . \quad (C-1)$$

As the droplet sweeps this volume, it encounters a number density of like droplets given by

$$N_m = \frac{1}{\frac{4}{3} \pi r_{pm}^3} \left(\frac{\alpha_{Lm}}{1 - \alpha_S} \right) \quad , \quad (C-2)$$

for a volume fraction α_{Lm} of liquid m. The ratio of $\alpha_{Lm}/(1 - \alpha_S)$, where α_S is the structure volume fraction, is the volume fraction of liquid m in the flow channel. As the liquid volume fraction approaches large values, the assumption of spherical droplets becomes invalid from overpacking. However, this means that the collision rate is large and coalescence is rapid.

The total number of collisions of this type per unit volume in the time interval Δt is the product of Eqs. (C-1) and (C-2) and the number of droplets that are sweeping individual volumes. Equation (C-3) gives this total frequency, or collisions per unit time and volume:

$$Z_m = \frac{9}{16\pi} \left(\frac{\alpha_{Lm}}{1 - \alpha_S} \right)^2 \frac{\bar{V}_m}{r_{pm}^4} \frac{1}{2} \quad . \quad (C-3)$$

Fig. C-1.
Physical concept for the droplet-droplet collision frequency.

The factor of one-half is included because each collision is counted twice. The rate of change of the droplet number density is directly proportional to the collision frequency as given by

$$\frac{dN_m}{dt} = -\omega Z_m \quad . \quad (C-4)$$

The minus sign is because the collisions reduce the number of droplets in the region. The proportionality constant, ω , is the coalescence probability per collision. Substitution of Eqs. (C-2) and (C-3) into Eq. (C-4) gives the rate of change of droplet radius:

$$\frac{dN_m}{dt} = -\frac{9}{4\pi} \left(\frac{\alpha_{Lm}}{1 - \alpha_S} \right) \frac{1}{r_{pm}^4} \frac{dr_{pm}}{dt} = -\omega \frac{9}{16\pi} \left(\frac{\alpha_{Lm}}{1 - \alpha_S} \right)^2 \frac{\bar{v}_m}{r_{pm}^4} \frac{1}{2} \quad , \quad \text{or} \quad (C-5)$$

$$\frac{dr_{pm}}{dt} = \frac{\omega \bar{v}_m}{8} \left(\frac{\alpha_{Lm}}{1 - \alpha_S} \right) = \text{COAL} \left(\frac{\alpha_{Lm}}{1 - \alpha_S} \right) \quad .$$

This expression easily is integrated over a time step to give the end-of-time-step radius, r_{pm}^{n+1} , based on the beginning-of-time-step radius, r_{pm}^n :

$$r_{pm}^{n+1} = \text{COAL} \left(\frac{\alpha_{Lm}^n}{1 - \alpha_S^n} \right) \Delta t + r_{pm}^n \quad , \quad (C-6)$$

where

$$\text{COAL} = \frac{\omega \bar{v}_m}{8} \quad . \quad (C-7)$$

APPENDIX D

LIQUID-LIQUID HEAT-TRANSFER MODEL

The liquid-liquid heat-transfer model is based on a droplet-droplet collision concept similar to that assumed for coalescence. The total collision rate per unit volume is derived in the following discussion along with the heat-transfer area, overall heat-transfer coefficient, and contact time per collision. The collision rate between liquid droplets of components m and k is formulated from the physical representation shown in Fig. D-1. Each droplet is assumed to have an average random velocity given by the difference between $\bar{V}_m$ and $\bar{V}_k$. These velocities differ from the liquid-field velocity. They are assumed to originate primarily from acceleration differences resulting from the different droplet sizes and densities. As a result, each droplet sweeps a volume of space per unit time given by the product of the random velocity and the projected area of the droplet. The volume swept per unit time and volume by all droplets or particles of component m is given by

$$VOL_m = n_m \pi r_{pm}^2 \bar{V}_m \quad , \quad (D-1)$$

where n_m is the number density of droplet component m . The number of collisions with droplets or particles of component k in VOL_m is simply VOL_m times the number density of type k . The collision rate per unit volume of component m with component k caused by the movement of component m is given by

$$Z_{m \rightarrow k} = \pi r_{pm}^2 \bar{V}_m n_m n_k \quad . \quad (D-2)$$

Similarly, the droplets of component k sweep a volume, resulting in collisions with type m . This rate is given by

$$Z_{k \rightarrow m} = \pi r_{pk}^2 \bar{V}_k n_m n_k \quad . \quad (D-3)$$

Fig. D-1.
Physical representation of liquid-liquid interaction.

The total collision rate is approximated by 0.5 times the sum of the rates given by Eqs. (D-2) and (D-3),

$$Z_{LmLk} = \pi n_m n_k (r_{pm}^2 \bar{v}_m + r_{pk}^2 \bar{v}_k) 0.5 \quad . \quad (D-4)$$

The factor of 0.5 accounts for counting each collision twice. The number density of droplets of each component is related to the respective volume fractions, α , and radii by

$$n_m = 3\alpha_{Lm}/4\pi r_{pm}^3 \quad \text{and} \quad (D-5)$$

$$n_k = 3\alpha_{Lk}/4\pi r_{pk}^3 \quad .$$

Substituting Eq. (D-5) into Eq. (D-4) gives the collision rate in terms of the variables computed by SIMMER-II in each mesh cell. Thus,

$$Z_{LmLk} = \frac{9}{32\pi} \frac{\alpha_{Lm}\alpha_{Lk}}{r_{pm}^3 r_{pk}^3} (r_{pm}^2 \bar{v}_m + r_{pk}^2 \bar{v}_k) \quad . \quad (D-6)$$

The collision rate does not behave well in the limit of the radii approaching zero. This situation is considered below in terms of physical limitations on the heat-transfer rates.

We consider the area of contact as the area for heat transfer that may be expected for each collision. Figure D-2 shows the assumed collision configuration for droplets of components m and k. In general, the contact area, A_{cLmLk} , may be highly variable because of its dependence on surface tension, velocity of impact, relative densities of the two components, and angle of impact. Though the mechanics of the impact may be complicated by the above factors, a reasonable assumption is that the contact area is on the order of the cross-section area of the smaller droplet. Therefore, we define the contact area to be

Fig. D-2.
Collision configuration for liquid m and liquid k .

$$A_{cLmLk} = C_1 \pi r_{pm}^2 \quad r_{pm} < r_{pk} \quad , \quad \text{and}$$

(D-7)

$$A_{cLmLk} = C_1 \pi r_{pk}^2 \quad r_{pm} \geq r_{pk} \quad .$$

The value of C_1 may range from 0.1 for a glancing collision or low-impact velocities to r_{pk}^2/r_{pm}^2 if $r_{pm} < r_{pk}$ or r_{pm}^2/r_{pk}^2 if $r_{pm} \geq r_{pk}$. This upper limit corresponds to the situation in which the smaller droplet deforms and covers an area equal to the cross-section area of the larger droplet.

The heat-transfer rate per unit area during the collision is assumed to be quasi-static. This is consistent with other heat-transfer modeling in SIMMER-II that approximates the process reasonably well for small droplets. In this particular case of spherical droplet interaction, it can be argued that because most of the heat capacity in a sphere is near the outer surface, the effective conduction path length is generally very small (about $0.2 r_p$). For droplets with radii of about 10^{-3} m, the thermal response time, τ , can be approximated by

$$\frac{\tau \alpha_t}{(0.2 r_p)^2} = 1 \quad , \quad (D-8)$$

where α_t is the thermal diffusivity of the liquid. Fuel, steel, and sodium droplets (10^{-3} m in radius) respond thermally in time intervals of 0.070 s, 0.008 s, and 0.0008 s respectively. Therefore, the quasi-static approximation is not good for fuel and steel droplets of 10^{-3} m radius. For short contact times and these relatively large droplets, the average heat-transfer rate during a collision will be underestimated substantially by this model.

The overall heat-transfer coefficient is based on conduction between two slabs shown in Fig. D-3 and assumed to have thicknesses of $0.2r_{pm}$ and $0.2r_{pk}$. These thicknesses represent the distance from the outer surface of a flattened or deformed sphere to the centroid surface. (One-half of the mass is inside this surface.) The overall heat-transfer coefficient is approximated by

Fig. D-3.
Assumed contact interval.

$$U_{cLmLk} = \frac{1}{\frac{0.2 r_{pm}}{k_{Lm}} + \frac{0.2 r_{pk}}{k_{Lk}}} \quad (D-9)$$

The contact time per collision also is uncertain because of the nature of the collision; that is, the velocity of collision, the angle of collision, the surface tension, and the thermal interaction at the contact surface (film boiling, and so on). We conveniently estimate the contact time on the basis of a mutual residence time considered to be proportional to the time that the two interacting droplets or particles are in possible contact. Figure D-4 illustrates the situation in which droplet k is moving relative to droplet m with an average velocity $\bar{V}$. The contact could begin when droplet k is at position A and end when it is at position B. The distance droplet k travels is about $2r_{pk} + 2r_{pm}$. The contact time, t_{cLmLk} , is given by

$$t_{cLmLk} = \frac{2C_2 (r_{pm} + r_{pk})}{\bar{V}/2} \quad (D-10)$$

The constant C_2 represents the fraction of the residence time that would characterize the average collision. The average velocity is divided by two because droplet m generally will be moving, and as a result, the relative velocity will be reduced.

The uncertainty in the average contact time is related to the uncertainty in the heat transfer per collision because of the thermal time constant considerations discussed previously. To some extent the uncertainties cancel. The error in the heat transfer rate during a collision, q_{cLmLk} , will be inversely proportional to the contact time because of the transient temperature gradient effect. Because the total heat transferred per collision is the product of q_{cLmLk} and t_{cLmLk} , the magnitude of the error in the total heat transferred may be considerably less than the individual errors in q_{cLmLk} and t_{cLmLk} .

The overall heat-transfer rate per unit volume is given by the product of the heat transferred per collision and the number of collisions per unit volume and time,

$$q_{HLmLk} = U_{cLmLk} A_{cLmLk} t_{cLmLk} (T_{Lk} - T_{Lm}) Z_{LmLk} \quad (D-11)$$

Fig. D-4.
Heat-transfer treatment during contact.

Substituting Eqs. (D-6), (D-7), (D-9), and (D-10) into (D-11) gives the heat-transfer rate,

$$q_{HLmLk} = C \frac{45}{8} \frac{1}{r_{pm}/k_{Lm} + r_{pk}/k_{Lk}} \frac{\alpha_{Lm} \alpha_{Lk}}{(1 - \alpha_S)} \frac{(r_{pm}^2 + r_{pk}^2)}{r_{pm}^3 r_{pk}^3} \cdot (r_{pm} + r_{pk}) r_c^2 (T_{Lk} - T_{Lm}) \quad , \quad (D-12)$$

where

$$r_c = r_{pk} \text{ if } r_{pk} < r_{pm},$$

$$r_c = r_{pm} \text{ if } r_{pk} > r_{pm}, \text{ and}$$

$$C = C_1 C_2 .$$

In Eq. (D-12), $\bar{V}_m$ and $\bar{V}_k$ have been approximated by $\bar{V}$. Thus, the random velocities of various types of droplets or particles are characterized by a single average value $\bar{V}$ that cancels in Eq. (D-12). The $(1 - \alpha_S)$ term because of structure is the net result of converting the liquid-field volume fractions from a total cell-volume basis to a flow-channel basis but keeping the heat-transfer rate on a total cell-volume basis, appropriate to its definition in the conservation equations.

As the radius, r_{pm} , of one of the droplets approaches zero, Eq. (D-12) reduces to

$$q_{HLmLk} \approx C \frac{45}{8} \frac{k_{Lk}}{r_{pk}} \frac{\alpha_{Lm} \alpha_{Lk}}{(1 - \alpha_S)} \frac{1}{r_{pm}} (T_{Lk} - T_{Lm}) \quad . \quad (D-13)$$

The heat-transfer rate approaches infinity as r_{pm} approaches zero. This results from the product of the number density of component m (Eq. D-5) and the contact area per collision, πr_{pm}^2 . Physically, the average instantaneous contact area exceeds the total surface area of all droplets of component k. This effect gives a heat-transfer rate greater than that which could occur with zero thermal resistance in the liquid component m. The limiting heat-transfer rate for this latter case is given by

$$(q_{HLmLk})_{\max} = 15 C \alpha_{Lk} \frac{k_{Lk}}{r_{pk}^2} (T_{Lk} - T_{Lm}) \text{ as } r_{pm} \text{ approaches zero} \quad , \quad (D-14)$$

or if r_{pk} approaches zero,

$$(q_{HLmLk})_{\max} = 15 C \alpha_{Lm} \frac{k_{Lm}}{r_{pm}^2} (T_{Lk} - T_{Lm}) \quad . \quad (D-15)$$

APPENDIX E

SPECIAL CASES IN THE PHASE-TRANSITION MODEL

The special procedures used to make the phase-transition model work under limiting conditions are detailed in this appendix. Based on the algorithm summarized at the end of Sec. IV.C.7, these procedures are divided into five categories as follows.

- (1) Adjustments to the heat-transfer coefficients during initialization of the model
- (2) Special cases within the inner iteration involving the saturation temperature and the mass conservation equations
- (3) Special cases in correcting the vapor-side heat-transfer coefficient based on the mass-transfer rates
- (4) Considerations in solving the vapor energy equations
- (5) Considerations with respect to the overall iteration procedure

First, because of the nonphysical results that can be obtained when heat-transfer coefficients are used in regimes beyond the validity of the programmed algorithms, the product of the heat-transfer coefficient and the area, H , can be adjusted each time step during model initialization. These adjustments are as follows.

- (a) If a material species is predicted to participate in the phase-transition model, the minimum product of the surface area and the vapor-side heat-transfer coefficient, H_{GLm} , is $10^{-6} \text{ W/(m}^2 \cdot \text{K)}$. This limit has been inserted primarily to avoid the problems of dividing by zero.
- (b) As expressed by Eq. (IV-89), the liquid-side coefficient, H_{LGm} , is limited so that heat transfer can be only 90% of $(\bar{\rho}_{Lm} c_{vLM})/\Delta\tilde{t}$. A constant liquid heat capacity is assumed below the liquid critical temperature because of the approximate nature of this formula. When $\hat{T}_{Lm} > T_{\text{Crt},M}$, the liquid heat capacity is given by c'_{vLM} , which is defined in Eq. (III-40). In addition, whenever Eq. (IV-89) does reduce the liquid-side coefficient, H_{LGm} , the vapor-side coefficient for the first iteration, H_{GLm}^1 , is determined by

$$H_{GLm}^1 = H_{GLm}^0 \frac{H_{LGm}}{H_{LGm}^0} \quad , \quad (E-1)$$

where H_{GLm}^0 is the product of the surface area and the initially computed droplet vapor-side heat-transfer coefficient for component m and

H_{LGm}^0 is the product of the surface area and the initially computed droplet liquid-side heat-transfer coefficient for component m.

- (c) As temperatures approach and/or exceed the critical temperature, the liquid-side heat-transfer coefficient should approach the vapor-side coefficient. The algorithms for computing heat-transfer coefficients do not possess this property. Consequently, the following arbitrary choices are made.

$$H_{GLm}^1 = H_{LGm} \quad \text{if } \hat{T}_{Lm} > 0.95 T_{Crt,M} \quad , \quad (E-2)$$

$$H_{GLm}^1 = \min [H_{LGm}, H_{GLm}^0 + (H_{LGm} - H_{GLm}^0) (T_G - T_{Crt,M} - 2000) * 2 \times 10^{-4}]$$

$$\text{if } \hat{T}_{Lm} \leq 0.95 T_{Crt,M} \quad , \quad (E-3)$$

$$T_{Sat,M}^n > T_{Crt,M} \quad , \quad \text{and}$$

$$T_G^n > T_{Crt,M} + 2000 \text{ K} \quad ,$$

where $T_{Sat,M}^n$ and T_G^n are values for the saturation temperature and vapor temperature determined at the beginning of the time step by the equation of state and H_{GLm}^0 in Eq. (E-3) could have been modified by Eq. (E-1) when the conditions for evaluating Eq. (E-1) are satisfied.

- (d) If supersaturated vapor is condensing, the vapor-side heat-transfer coefficient will be increased because of mass transfer according to Eq. (IV-103) or its analog for a structure surface. However, if highly supersaturated conditions exist, the equation of state sometimes will require an increase in the vapor-specific internal energy to maintain a reasonable vapor temperature. To obtain an appropriate energy addition to the vapor, the vapor-side heat-transfer coefficient must be increased. Highly supersaturated conditions are considered to exist if

the vapor saturation temperature exceeds both the vapor temperature and the temperature of the liquid-field component by 300 K, that is,

$$T_{\text{Sat},M}^n > T_G^1 + 300 \quad \text{and}$$

$$T_{\text{Sat},M}^n > \hat{T}_{Lm} + 300 \quad . \quad (\text{E-4})$$

If both these conditions are satisfied, the vapor-side heat-transfer coefficient multiplied by the appropriate surface area is given by

$$H_{GLm}^1 = \max (H_{GLm}^0, H_{LGm}^1) \quad . \quad (\text{E-5})$$

Here, T_G^1 , the vapor temperature for iteration 1, may differ from T_G^0 through use of Eq. (E-22) below, and H_{GLm}^0 may have been modified previously by Eq. (E-1), Eq. (E-2), or Eq. (E-3).

- (e) If H_{GLm}^1 is allowed to differ from H_{GLm}^0 through the above relationships, H_{GLm}^* is adjusted to agree with H_{GLm}^1 rather than H_{GLm}^0 . The mass-transfer modifications to the vapor-side heat-transfer coefficient calculated by Eq. (IV-104) then should be consistent with these preliminary adjustments.

Second, we must consider the inner iteration involving the equation of state and the mass conservation equations. Section IV describes the procedure if ρ_{Gm}^{k+1} exceeds the limitations set by the possible amount of material. The remaining problem concerns oscillations when a component's partial pressure is near the critical pressure. The crucial term is

$$\frac{1}{h_{lg,M}^k} (a_{1,m}^k + a_{2,m}^k T_{\text{Sat},M}^k)$$

in Eq. (IV-100). Here, $a_{2,m}^k < 0$, whereas $a_{1,m}^k > 0$. Therefore, if $(-a_{1,m}^k/a_{2,m}^k) > T_{\text{Crt},M}$, the above term will be positive for all $T_{\text{Sat},M}^k < T_{\text{Crt},M}$. However, if $(-a_{1,m}^k/a_{2,m}^k) < T_{\text{Crt},M}$, the above term will become negative for some $T_{\text{Sat},M}^k < T_{\text{Crt},M}$. This property allows the function F to vary rapidly near the critical pressure because of the rapid change in $h_{lg,M}^k$. For example, as com-

monly fitted, $h_{lg,M}$ in the analytical equation of state for sodium varies from 3.3×10^5 J/kg at 1 K below the critical temperature to 100 J/kg (the minimum allowed) at the critical temperature. An example of a possible F function (whose zero must be found) as evaluated for water in the Zion/Indian-Point study¹ is shown in Figs. E-1 and E-2. Obviously, an approach to the solution cannot be obtained unless derivatives are taken sufficiently close to the point of solution. If oscillations of $T_{Sat,M}^k$ above and below the critical temperature are detected in successive inner iterations and if $(-a_{1,m}^k/a_{2,m}^k)$ is less than $T_{Crt,M}$, a special procedure is used to nudge the iteration procedure towards convergence. Starting with $T_{Sat,M}^k > T_{Crt,M}$ and $T_{Sat,M}^{k-1} < T_{Crt,M}$, the variables from the k iteration are redefined by

$$T_{Sat,M}^k = \frac{1}{2} (T_{Sat,M}^{k-1} + T_{Crt,M}) \quad , \quad (E-6)$$

$$\bar{p}_{Gm}^k = \min [\bar{p}_{Gm}^k, \bar{p}_{Gm}^{k-1}] \quad , \quad (E-7)$$

$$h_{lg,M}^k = \frac{1}{2} [h_{lg,M}^{k-1} + \frac{a_{1,m}^k + a_{2,m}^k T_{Crt,M}}{(\bar{p}_{Gm}^{k-1} - \bar{p}_{Gm}^n)}] \quad , \quad (E-8)$$

and

$$\left(\frac{\partial h_{lg,M}}{\partial T_{Sat,M}} \right)^k = \frac{-h_{lg,M}^k}{(T_{Crt,M} - T_{Sat,M}^k)} \quad , \quad (E-9)$$

where the updated $h_{lg,M}^k$ and $T_{Sat,M}^k$ from Eqs. (E-8) and (E-6) are used in Eq. (E-9). If $T_{Sat,M}^k$ from Eq. (E-6) remains greater than $T_{Crt,M}$, the resets are repeated. Eventually, $T_{Sat,M}^k < T_{Crt,M}$. However, in general, the value of $T_{Sat,M}^{k+1}$ calculated after these redefinitions [from an equation-of-state call

Fig. E-1.
Plot of F as a function of T_{Sat} from Eq. (IV-100) for a typical situation with convergence difficulties.

Fig. E-2.
Enlargement of the F function in the vicinity of the critical temperature.

following evaluation of Eq. (IV-101)] will not be sufficiently close to the converged value of the saturation temperature to permit the iteration to converge. To limit the possible overshoot of $T_{\text{Sat},M}^{\hat{k}+2}$, $h_{\ell g,M}^{\hat{k}+1}$ is adjusted according to

$$(h_{\ell g,M}^{\hat{k}+1})_{\text{reset}} = 0.9 h_{\ell g,M}^{\hat{k}} + 0.1 h_{\ell g,M}^{\hat{k}+1}, \quad \text{if } h_{\ell g,M}^{\hat{k}+1} < 10 h_{\ell g,M}^{\hat{k}}, \quad (\text{E-10})$$

or

$$(h_{\ell g,M}^{\hat{k}+1})_{\text{reset}} = 0.9 h_{\ell g,M}^{\hat{k}} + 0.02 h_{\ell g,M}^{\hat{k}+1}, \quad \text{if } h_{\ell g,M}^{\hat{k}+1} \geq 10 h_{\ell g,M}^{\hat{k}}.$$

Further, if $a_{1,m}^{\kappa} + a_{2,m}^{\kappa} T_{\text{Sat},M}^{\hat{k}+1} > 0$, a large negative $(\frac{\partial h_{\ell g,M}}{\partial T_{\text{Sat},M}})^{\hat{k}+1}$

can cause divergence (unless $T_{\text{Sat},M}^{\hat{k}+1}$ is near the solution, which is not guaranteed). Consequently, if

$$a_{1,m}^{\kappa} + a_{2,m}^{\kappa} T_{\text{Sat},m}^{\hat{k}+1} > 0, \text{ the term } (\frac{\partial h_{\ell g,M}}{\partial T_{\text{Sat},M}})^{\hat{k}+1} \text{ is set to } -0.001.$$

This entire reset procedure depends on detecting of oscillations at the earliest possible inner iteration so that the number of resets can be reduced when calculating successive outer iterations. If the initial saturation temperature is above the critical temperature, oscillations can be detected after the third iteration. For initial saturation temperatures below the critical temperature and $a_{1,m}^{\kappa} < a_{1,m}^{\kappa-1}$, convergence above the critical temperature is impossible and oscillation tests can start after the first iteration. If $a_{1,m}^{\kappa}$ has increased, the situation is more complex, and iteration tests are started after the third iteration if $T_G^{\kappa} < T_G^{\kappa-1}$ (in other words, if the increase in $a_{1,m}^{\kappa}$ is a result of the heat-transfer coefficient). Otherwise, iteration tests are begun after the second iteration.

Finally, difficulties can arise even if $(-a_{1,m}^{\kappa}/a_{2,m}^{\kappa}) > T_{\text{Crt},M}$. In this case, convergence below the critical temperature is judged to be unlikely if oscillations start or if $\hat{T}_{\text{Lm}} > T_{\text{Crt},M}$. In this situation, $h_{\ell g,M}^{\hat{k}}$ is set to 100 J/kg, and

the term $(\frac{\partial h_{g,M}}{\partial T_{Sat,M}})^k$ is set to -0.001.

This case does allow vaporization and condensation. However, given the small vapor-field energy change, the final pressure is essentially the liquid-vapor pressure corresponding to equilibrium (assuming that enough liquid is available). If convergence does not occur, an additional flag is set to initiate an error exit from the vaporization-condensation model.

The FORTRAN indices used in these special cases are given in Table E-I. If future versions of SIMMER-II are developed, a bisection solution algorithm will be formulated to find a zero for Eq. (IV-100) in domains of convergence difficulties.

TABLE E-I
INDICES USED FOR THE INNER ITERATION ON THE
CONSERVATION EQUATIONS AND THE SATURATION TEMPERATURE

<u>Variable</u>	<u>Value</u>	<u>Definition</u>
IRHOIT	0-100	Iteration index
ITSP(N)	1-3	Indicates the number of possible iterations before oscillations can be detected in material N.
	4	Used to transfer to a quasi-equilibrium model not available in the current version of SIMMER-II.
IVPALL(N)	0	Nominal Case
	1	Indicates all the liquid of material N is vaporizing.
ICOUNT(N)	0	Nominal Case
	1	Indicates a reset was done on the previous iteration on material N.
		Necessary for changing $h_{g,M}^{k+1}$, and $(\frac{\partial h_{g,M}}{\partial T_{sat,M}})^{k+1}$.
	3	Used when convergence below the critical temperature is judged to be unlikely.

Third, the expression used for the Newton-Raphson procedure for correcting the vapor-side heat-transfer coefficients based on the mass-transfer rates at the surface of a drop of material m is

$$F = \frac{\bar{c}_{pG}}{h_{\ell g, M}} \{H_{LGm}(T_{Sat, M} - \hat{T}_{Lm}) + \frac{x}{e^{x/H_{GLm}^*}} (T_{Sat, M} - T_G)\} + x = 0 \quad , \quad (E-11)$$

where $x = -\bar{c}_{pG} \Gamma_{IGmLm}$.

The iteration is described by

$$x^{k+1} = x^k - F^k / \left(\frac{dF}{dx} \right)^k \quad . \quad (E-12)$$

The vapor-side heat-transfer coefficient is given by Eq. (IV-103) following convergence of Eq. (E-11) to six significant figures. The vapor-side heat-transfer coefficient at a structure surface is obtained from a similar iteration by evaluating F from a summation over all the components exchanging mass at the interface.

The special cases for this iteration are as follows.

- (a) If $\hat{T}_{Lm}$ is greater than $0.95 T_{Crt, M}$, the quantity $\bar{c}_{pG}/h_{\ell g, M}$ in Eq. (E-11) is set to zero. At these temperatures, phenomena such as homogeneous nucleation become possible, and the mass-transfer rate is no longer presumed to influence the heat-transfer coefficient. This means that $H_{GLm} = H_{GLm}^*$.
- (b) If H_{GLm}^* is less than $1 \text{ W/m}^2 \cdot \text{K}$, revision of the heat-transfer coefficient is not attempted.
- (c) If $T_{Sat, M}^n$ is greater than $T_{Crt, M}$, vaporization is not allowed to decrease the vapor-side heat-transfer coefficient.
- (d) If x/H_{GLm}^* is small (having an absolute value less than 10^{-3} initially or less than 10^{-4} following any iteration), the exponential in Eq. (IV-103) is approximated by a quadratic expansion, and H_{GLm} is evaluated directly.

(evaluated using the higher temperature solution) is more than 2 000 K above any beginning-of-iteration structure, liquid, or vapor temperature in the cell.

The remaining special cases for the solution to the vapor energy equation are as follows.

- (a) If the coefficients of the quadratic equation [Eq. (IV-113)] are such that

$$B^2 - 4AC < 0 \quad , \quad (E-15)$$

no real solution is possible. In such a case, the vapor temperature is estimated to be

$$T_G^{k+1} = \frac{-B}{2A} \quad . \quad (E-16)$$

If conditions are satisfied for convergence of the outer phase-transition iteration while this condition exists, the entire phase-transition calculation is redone using the lower temperature branch of the vapor energy equation (when one exists). If the outer iteration still converges with no solution possible, the model error exit is taken.

- (b) If the coefficient on the quadratic term, A, in Eq. (IV-113) is small such that $|A| < 10^{-8} |B|$, then the vapor temperature is defined from

$$T_G^{k+1} = \frac{-C}{B} \quad . \quad (E-17)$$

- (c) If T_G^{k+1} is less than zero, the vapor temperature is redefined from

$$T_G^{k+1} = \frac{T_G^k}{2} \quad . \quad (E-18)$$

(evaluated using the higher temperature solution) is more than 2 000 K above any beginning-of-iteration structure, liquid, or vapor temperature in the cell.

The remaining special cases for the solution to the vapor energy equation are as follows.

- (a) If the coefficients of the quadratic equation [Eq. (IV-113)] are such that

$$B^2 - 4AC < 0 \quad , \quad (E-15)$$

no real solution is possible. In such a case, the vapor temperature is estimated to be

$$T_G^{k+1} = \frac{-B}{2A} \quad . \quad (E-16)$$

If conditions are satisfied for convergence of the outer phase-transition iteration while this condition exists, the entire phase-transition calculation is redone using the lower temperature branch of the vapor energy equation (when one exists). If the outer iteration still converges with no solution possible, the model error exit is taken.

- (b) If the coefficient on the quadratic term, A, in Eq. (IV-113) is small such that $|A| < 10^{-8} |B|$, then the vapor temperature is defined from

$$T_G^{k+1} = \frac{-C}{B} \quad . \quad (E-17)$$

- (c) If T_G^{k+1} is less than zero, the vapor temperature is redefined from

$$T_G^{k+1} = \frac{T_G^k}{2} \quad . \quad (E-18)$$

Fifth, the special considerations for the overall iteration are as follows.

- (a) If the vaporization-condensation calculation does not converge within 20 outer iterations, the heat-transfer coefficient iteration is terminated. Upon termination of this inner iteration for a given time step, the vapor-side heat-transfer coefficients for the droplet components are reset to the maximum of four values consisting of (1) the initially computed value at iteration 1 and (2) one of the stored values from iteration 18, 19, or 20. The vapor-side heat-transfer coefficient for structure is allowed to remain unchanged (at the last iterative value) unless $\hat{T}_{Lm} \geq T_{Crt,M}$. In this case, the same prescription is used as in the droplet situation.
- (b) If a vapor-field component saturation temperature is less than any structure-field component temperature and only a small amount of liquid of the vapor-field component exists, structure-to-vapor heat transfer is calculated directly rather than through vaporization of the small amount of remaining liquid. Explicitly, these controlling conditions are $T_{Sat,M}^n \leq T_{Sk}^n$ and $\alpha_{Lm} \leq \alpha_o(1 - \alpha_S)$. The product of the surface area and heat-transfer coefficient, $\hat{H}_{GSk}$, for the vapor-structure heat transfer is defined as

$$\frac{1}{\hat{H}_{GSk}} = \frac{1}{H_{SGk}} + \frac{1}{H_{Gsk}} \quad , \quad (E-19)$$

where H_{SGk} is the usual product of the structure surface area and the structure-side heat-transfer coefficient and H_{Gsk} is the usual product of the structure surface area and the vapor-side heat-transfer coefficient to structure. The heat transfer is evaluated implicitly starting from the truncated energy equation for the vapor field:

$$\frac{\partial(\bar{p}_G \bar{c}_{vG} T_G)}{\partial t} = \hat{H}_{GSk} (\tilde{T}_{Sk}^{n+1} - T_G^{n+1}) \quad . \quad (E-20)$$

This yields

$$\Delta T_G = [\Delta \tilde{t} \hat{H}_{GSk} (\tilde{T}_{Sk}^{n+1} - \tilde{T}_G)] / (\bar{\rho}_G \bar{c}_{vG} + \Delta t \hat{H}_{GSk}) \quad , \quad (E-21)$$

where $\tilde{T}_G$ is the final vapor-field temperature following the evaluation of the vaporization and condensation rates. The actual energy-transfer rate then is

$$(\bar{\rho}_G \bar{c}_{vG} \frac{\Delta T_G}{\Delta \tilde{t}}) \quad .$$

- (c) Convergence can be improved for the case of high liquid temperatures if the initial estimate of T_G is near the anticipated converged value. The prescription to revise the initial estimate to T_G is

$$T_G^1 = \min [\hat{T}_{Lm}, \max (T_{Crt,M}, T_G^0)], \quad \text{if } \hat{T}_{Lm} > 0.95 T_{Crt,M} \quad , \quad (E-22)$$

where T_G^0 is the initially computed vapor temperature estimate from the equation of state.*

- (d) If the iteration involving recalculating the vapor-side heat-transfer coefficient predicts an excessive increase over the initial value, oscillations between this calculation and the vapor-field energy equation may occur. Because the presence of noncondensable species will tend to restrict the magnitude of this increase, such a situation may not be realistic. Thus, the magnitudes of the vapor-side heat-transfer coefficients for both liquid-field and structure-field components are restricted to twice their respective liquid-side and structure-side values. Fortunately, the liquid-side and structure-side heat-transfer coefficients are generally more than an order of magnitude above the vapor-side values.

*This modification is written in terms of one component only because there is insufficient experience for the case of multiple components, all with $T_{Lm} > T_{Crt,M}$.

- (e) To avoid oscillations resulting from special termination conditions, for example, all the liquid vaporizing, limitations are set on the final two variables in Table E-I. After the tenth outer iteration, if all of a component is indicated to vaporize, this component must stay vaporized in subsequent iterations [IVPALL(N) will remain at 1]. Also, if ICOUNT(N) is set to 3, indicating that $T_{Sat,M}$ should be greater than $T_{Crt,M}$, a reset of ICOUNT(N) to zero will occur only before the first four outer iterations and at the beginning of the tenth outer iteration.
- (f) The variable ICOUNT(N) in Table E-I is set to 3 if $h_{lg,M} = 100$ J/kg (the minimum allowed) and the number of outer iterations is greater than 20. This eliminates the possibility of increasing the heat of vaporization for this component during this time step.
- (g) If ICOUNT(N) is 3, the energies of the saturated vapor and saturated liquid for a component are adjusted for consistency if necessary. Here the equation of state may not necessarily be calculating an above-critical-point saturation temperature consistent with the negligible heat of vaporization used.

REFERENCES

1. M. G. Stevenson, "Report of the Zion/Indian Point Study," Volume II, Los Alamos National Laboratory LA-8306-MS, NUREG/CR-1411 report (April 1980).

APPENDIX F

THE PRESSURE ITERATION MATRIX

Because the mass-continuity equation residual $\tilde{D}$ is a function of the local mesh cell pressure and the adjacent mesh cell pressures, $\tilde{D}$ for iteration $\kappa+1$ can be written as a Taylor series expansion about iteration κ truncated to first order:

$$\begin{aligned} \tilde{D}_{ij}^{\kappa+1} = \tilde{D}_{ij}^{\kappa} &+ \left[\frac{\partial \tilde{D}_{ij}^{\kappa}}{\partial p_{ij}} \right] \delta p_{ij} + \left[\frac{\partial \tilde{D}_{ij}^{\kappa}}{\partial p_{i+1,j}} \right] \delta p_{i+1,j} + \left[\frac{\partial \tilde{D}_{ij}^{\kappa}}{\partial p_{i-1,j}} \right] \delta p_{i-1,j} \\ &+ \left[\frac{\partial \tilde{D}_{ij}^{\kappa}}{\partial p_{i,j+1}} \right] \delta p_{i,j+1} + \left[\frac{\partial \tilde{D}_{ij}^{\kappa}}{\partial p_{i,j-1}} \right] \delta p_{i,j-1} \quad . \end{aligned} \quad (F-1)$$

For two-phase cells, $\tilde{D}$ is obtained from the vapor continuity equation as

$$\tilde{D}_G^{\kappa} = \frac{1}{\Delta t} [(\tilde{\rho}_G)^{\kappa} - (\rho_G)^n] + \frac{\Delta_i < \tilde{\rho}_G^{\kappa} r^{\frac{1}{2}} \tilde{U}_G^{\kappa} >_j}{r_i^{\frac{1}{2}} \Delta r_i} + \frac{\Delta_j < \tilde{\rho}_G^{\kappa} \tilde{v}_G^{\kappa} >_i}{\Delta z_j} + \Gamma_G \quad . \quad (F-2)$$

The pressure derivatives are

$$\begin{aligned} \frac{\partial \tilde{D}_{Gij}^{\kappa}}{\partial p_{ij}} = \frac{1}{\Delta t} \left(\frac{\partial \tilde{\rho}_G^{\kappa}}{\partial p} \right)_{ij} &+ \frac{1}{r_i^{\frac{1}{2}} \Delta r_i} \left[\frac{\partial < \tilde{\rho}_G^{\kappa} r^{\frac{1}{2}} \tilde{U}_G^{\kappa} >_{i+1/2,j}}{\partial p_{ij}} - \frac{\partial < \tilde{\rho}_G^{\kappa} r^{\frac{1}{2}} \tilde{U}_G^{\kappa} >_{i-1/2,j}}{\partial p_{ij}} \right] \\ &+ \frac{1}{\Delta z_j} \left[\frac{\partial < \tilde{\rho}_G^{\kappa} \tilde{v}_G^{\kappa} >_{i,j+1/2}}{\partial p_{ij}} - \frac{\partial < \tilde{\rho}_G^{\kappa} \tilde{v}_G^{\kappa} >_{i,j-1/2}}{\partial p_{ij}} \right] \quad , \end{aligned} \quad (F-3)$$

$$\frac{\partial \tilde{D}_{Gij}^{\kappa}}{\partial p_{i+1,j}} = \frac{1}{r_i^{\kappa} \Delta r_i} \left[\frac{\partial < \tilde{\rho}_G^{\kappa} r^{\kappa} \tilde{U}_G^{\kappa} >_{i+1/2,j}}{\partial p_{i+1,j}} \right] , \quad (F-4)$$

$$\frac{\partial \tilde{D}_{Gij}^{\kappa}}{\partial p_{i-1,j}} = \frac{-1}{r_i^{\kappa} \Delta r_i} \left[\frac{\partial < \tilde{\rho}_G^{\kappa} r^{\kappa} \tilde{U}_G^{\kappa} >_{i-1/2,j}}{\partial p_{i-1,j}} \right] , \quad (F-5)$$

$$\frac{\partial \tilde{D}_{Gij}^{\kappa}}{\partial p_{i,j+1}} = \frac{1}{\Delta z_j} \left[\frac{\partial < \tilde{\rho}_G^{\kappa} \tilde{V}_G^{\kappa} >_{i,j+1/2}}{\partial p_{i,j+1}} \right] , \quad \text{and} \quad (F-6)$$

$$\frac{\partial \tilde{D}_{Gij}^{\kappa}}{\partial p_{i,j-1}} = \frac{-1}{\Delta z_j} \left[\frac{\partial < \tilde{\rho}_G^{\kappa} \tilde{V}_G^{\kappa} >_{i,j-1/2}}{\partial p_{i,j-1}} \right] . \quad (F-7)$$

The derivatives themselves are evaluated using donor-cell differencing, where the explicit representation for the four repeated expressions in the broken brackets is

$$\begin{aligned} < \tilde{\rho}_G^{\kappa} r^{\kappa} \tilde{U}_G^{\kappa} >_{i+1/2,j} = (r^{\kappa} \tilde{U}_G^{\kappa})_{i+1/2,j} \left[\left(\frac{\Delta r_{i+1}}{2\Delta r_{i+1/2}} + \eta_{i+1/2} \right) \tilde{\rho}_{Gij}^{\kappa} \right. \\ \left. + \left(\frac{\Delta r_i}{2\Delta r_{i+1/2}} - \eta_{i+1/2} \right) \tilde{\rho}_{G_{i+1,j}}^{\kappa} \right] , \end{aligned} \quad (F-8)$$

$$\begin{aligned} < \tilde{\rho}_G^{\kappa} r^{\kappa} \tilde{U}_G^{\kappa} >_{i-1/2,j} = (r^{\kappa} \tilde{U}_G^{\kappa})_{i-1/2,j} \left[\left(\frac{\Delta r_i}{2\Delta r_{i-1/2}} + \eta_{i-1/2} \right) \tilde{\rho}_{G_{i-1,j}}^{\kappa} \right. \\ \left. + \left(\frac{\Delta r_{i-1}}{2\Delta r_{i-1/2}} - \eta_{i-1/2} \right) \tilde{\rho}_{Gij}^{\kappa} \right] , \end{aligned} \quad (F-9)$$

$$\begin{aligned}
\langle \tilde{\rho}_G^\kappa \tilde{v}_G^\kappa \rangle_{i,j+1/2} &= \tilde{v}_{Gi,j+1/2}^\kappa \left[\left(\frac{\Delta z_{j+1}}{2\Delta z_{j+1/2}} + \eta_{j+1/2} \right) \tilde{\rho}_{Gi,j}^\kappa \right. \\
&\quad \left. + \left(\frac{\Delta z_j}{2\Delta z_{j+1/2}} - \eta_{j+1/2} \right) \tilde{\rho}_{Gi,j+1}^\kappa \right] ,
\end{aligned} \tag{F-10}$$

and

$$\begin{aligned}
\langle \tilde{\rho}_G^\kappa \tilde{v}_G^\kappa \rangle_{i,j-1/2} &= \tilde{v}_{Gi,j-1/2}^\kappa \left[\left(\frac{\Delta z_j}{2\Delta z_{j-1/2}} + \eta_{j-1/2} \right) \tilde{\rho}_{Gi,j-1}^\kappa \right. \\
&\quad \left. + \left(\frac{\Delta z_{j-1}}{2\Delta z_{j-1/2}} - \eta_{j-1/2} \right) \tilde{\rho}_{Gi,j}^\kappa \right] ,
\end{aligned} \tag{F-11}$$

where

$$\eta_{i+1/2} = \frac{a_0}{\Delta r_{i+1/2}} \left[\begin{array}{ll} \Delta r_i & , \text{ if } \tilde{U}_{Gi+1/2,j}^\kappa \geq 0 \\ -\Delta r_{i+1} & , \text{ if } \tilde{U}_{Gi+1/2,j}^\kappa < 0 \end{array} \right] + \frac{b_0 \Delta t}{\Delta r_{i+1/2}} \tilde{U}_{Gi+1/2,j}^\kappa ,$$

$$\eta_{i-1/2} = \frac{a_0}{\Delta r_{i-1/2}} \left[\begin{array}{ll} \Delta r_{i-1} & , \text{ if } \tilde{U}_{Gi-1/2,j}^\kappa \geq 0 \\ -\Delta r_i & , \text{ if } \tilde{U}_{Gi-1/2,j}^\kappa < 0 \end{array} \right] + \frac{b_0 \Delta t}{\Delta r_{i-1/2}} \tilde{U}_{Gi-1/2,j}^\kappa ,$$

$$\eta_{j+1/2} = \frac{a_0}{\Delta z_{j+1/2}} \left[\begin{array}{ll} \Delta z_j & , \text{ if } \tilde{v}_{Gi,j+1/2}^\kappa \geq 0 \\ -\Delta z_{j+1} & , \text{ if } \tilde{v}_{Gi,j+1/2}^\kappa < 0 \end{array} \right] + \frac{b_0 \Delta t}{\Delta z_{j+1/2}} \tilde{v}_{Gi,j+1/2}^\kappa , \text{ and}$$

$$\eta_{j-1/2} = \frac{a_0}{\Delta z_{j-1/2}} \left[\begin{array}{ll} \Delta z_{j-1} & , \text{ if } \tilde{v}_{Gi,j-1/2}^\kappa \geq 0 \\ -\Delta z_j & , \text{ if } \tilde{v}_{Gi,j-1/2}^\kappa < 0 \end{array} \right] + \frac{b_0 \Delta t}{\Delta z_{j-1/2}} \tilde{v}_{Gi,j-1/2}^\kappa .$$

The quantities $\eta_{i+1/2}$, $\eta_{i-1/2}$, $\eta_{j+1/2}$, and $\eta_{j-1/2}$ are not differentiated with respect to pressure. Usually b_0 is input as zero, making the expressions inde-

pendent of pressure if the velocity components do not change sign. Hence, the derivatives of Eqs. (F-8) to (F-11) are

$$\frac{\partial < \tilde{\rho}_G^\kappa r^\xi \tilde{U}_G^\kappa >_{i+1/2, j}}{\partial p_{ij}} = r_{i+1/2}^\xi \{ \tilde{U}_{Gi+1/2, j}^\kappa [(\frac{\Delta r_{i+1}}{2\Delta r_{i+1/2}} + \eta_{i+1/2}) \frac{\partial \tilde{\rho}_{Gi j}^\kappa}{\partial p_{ij}}]$$

(F-12)

$$+ \frac{\partial \tilde{U}_{Gi+1/2, j}^\kappa}{\partial p_{ij}} [(\frac{\Delta r_{i+1}}{2\Delta r_{i+1/2}} + \eta_{i+1/2}) \tilde{\rho}_{Gi j}^\kappa + (\frac{\Delta r_i}{2\Delta r_{i+1/2}} - \eta_{i+1/2}) \tilde{\rho}_{Gi+1, j}^\kappa] \} ,$$

$$\frac{\partial < \tilde{\rho}_G^\kappa r^\xi \tilde{U}_G^\kappa >_{i-1/2, j}}{\partial p_{ij}} = r_{i-1/2}^\xi \{ \tilde{U}_{Gi-1/2, j}^\kappa [(\frac{\Delta r_{i-1}}{2\Delta r_{i-1/2}} - \eta_{i-1/2}) \frac{\partial \tilde{\rho}_{Gi j}^\kappa}{\partial p_{ij}}]$$

(F-13)

$$+ \frac{\partial \tilde{U}_{Gi-1/2, j}^\kappa}{\partial p_{ij}} [(\frac{\Delta r_i}{2\Delta r_{i-1/2}} + \eta_{i-1/2}) \tilde{\rho}_{Gi-1, j}^\kappa + (\frac{\Delta r_{i-1}}{2\Delta r_{i-1/2}} - \eta_{i-1/2}) \tilde{\rho}_{Gi j}^\kappa] \} ,$$

$$\frac{\partial < \tilde{\rho}_G^\kappa \tilde{V}_G^\kappa >_{i, j+1/2}}{\partial p_{ij}} = \tilde{V}_{Gi, j+1/2}^\kappa [(\frac{\Delta z_{j+1}}{2\Delta z_{j+1/2}} + \eta_{j+1/2}) \frac{\partial \tilde{\rho}_{Gi j}^\kappa}{\partial p_{ij}}]$$

(F-14)

$$+ \frac{\partial \tilde{V}_{Gi, j+1/2}^\kappa}{\partial p_{ij}} [(\frac{\Delta z_{i+1}}{2\Delta z_{j+1/2}} + \eta_{j+1/2}) \tilde{\rho}_{Gi j}^\kappa + (\frac{\Delta z_j}{2\Delta z_{j+1/2}} - \eta_{j+1/2}) \tilde{\rho}_{Gi, j+1}^\kappa] ,$$

$$\frac{\partial < \tilde{\rho}_G^\kappa \tilde{v}_G^\kappa >_{i,j-1/2}}{\partial p_{ij}} = \tilde{v}_{Gi,j-1/2}^\kappa \left[\left(\frac{\Delta z_{i-1}}{2\Delta z_{i-1/2}} - \eta_{j-1/2} \right) \frac{\partial \tilde{\rho}_{Gi,j}^\kappa}{\partial p_{ij}} \right]$$

(F-15)

$$+ \frac{\partial \tilde{v}_{Gi,j-1/2}^\kappa}{\partial p_{ij}} \left[\left(\frac{\Delta z_j}{2\Delta z_{j-1/2}} + \eta_{j-1/2} \right) \tilde{\rho}_{Gi,j-1}^\kappa + \left(\frac{\Delta z_{j-1}}{2\Delta z_{j-1/2}} - \eta_{j-1/2} \right) \tilde{\rho}_{Gi,j}^\kappa \right]$$

$$\frac{\partial < \tilde{\rho}_G^\kappa r^\xi \tilde{u}_G^\kappa >_{i+1/2,j}}{\partial p_{i+1,j}} = r_{i+1/2}^\xi \{ \tilde{u}_{Gi+1/2,j}^\kappa \left[\left(\frac{\Delta r_i}{2\Delta r_{i+1/2}} - \eta_{i+1/2} \right) \frac{\partial \tilde{\rho}_{Gi+1,j}^\kappa}{\partial p_{i+1,j}} \right] \right]$$

(F-16)

$$+ \frac{\partial \tilde{u}_{Gi+1/2,j}^\kappa}{\partial p_{i+1,j}} \left[\left(\frac{\Delta r_{i+1}}{2\Delta r_{i+1/2}} + \eta_{i+1/2} \right) \tilde{\rho}_{Gi,j}^\kappa + \left(\frac{\Delta r_i}{2\Delta r_{i+1/2}} - \eta_{i+1/2} \right) \tilde{\rho}_{Gi+1,j}^\kappa \right]$$

$$\frac{\partial < \tilde{\rho}_G^\kappa r^\xi \tilde{u}_G^\kappa >_{i-1/2,j}}{\partial p_{i-1,j}} = r_{i-1/2}^\xi \{ \tilde{u}_{Gi-1/2,j}^\kappa \left[\left(\frac{\Delta r_i}{2\Delta r_{i-1/2}} + \eta_{i+1/2} \right) \frac{\partial \tilde{\rho}_{Gi-1,j}^\kappa}{\partial p_{i-1,j}} \right] \right]$$

(F-17)

$$+ \frac{\partial \tilde{u}_{Gi-1/2,j}^\kappa}{\partial p_{i-1,j}} \left[\left(\frac{\Delta r_i}{2\Delta r_{i-1/2}} + \eta_{i-1/2} \right) \tilde{\rho}_{Gi-1,j}^\kappa + \left(\frac{\Delta r_{i-1}}{2\Delta r_{i-1/2}} - \eta_{i-1/2} \right) \tilde{\rho}_{Gi,j}^\kappa \right]$$

$$\frac{\partial < \tilde{\rho}_G^\kappa \tilde{v}_G^\kappa >_{i,j+1/2}}{\partial p_{i,j+1}} = \tilde{v}_{Gi,j+1/2}^\kappa \left[\left(\frac{\Delta z_j}{2\Delta z_{j+1/2}} - \eta_{j+1/2} \right) \frac{\partial \tilde{\rho}_{Gi,j+1}^\kappa}{\partial p_{i,j+1}} \right]$$

(F-18)

$$+ \frac{\partial \tilde{v}_{Gi,j+1/2}^{\kappa}}{\partial p_{i,j+1}} \left[\left(\frac{\Delta z_{j+1}}{2\Delta z_{j+1/2}} + \eta_{j+1/2} \right) \tilde{\rho}_{Gi,j}^{\kappa} + \left(\frac{\Delta z_j}{2\Delta z_{j+1/2}} - \eta_{j+1/2} \right) \tilde{\rho}_{Gi,j+1}^{\kappa} \right] , \text{ and}$$

$$\frac{\partial < \tilde{\rho}_G^{\kappa} \tilde{v}_G^{\kappa} >_{i,j-1/2}}{\partial p_{i,j-1}} = \tilde{v}_{Gi,j-1/2}^{\kappa} \left[\left(\frac{\Delta z_j}{2\Delta z_{j-1/2}} + \eta_{j-1/2} \right) \frac{\partial \tilde{\rho}_{Gi,j-1}^{\kappa}}{\partial p_{i,j-1}} \right]$$

(F-19)

$$+ \frac{\partial \tilde{v}_{Gi,j-1/2}^{\kappa}}{\partial p_{i,j-1}} \left[\left(\frac{\Delta z_j}{2\Delta z_{j-1/2}} + \eta_{j-1/2} \right) \tilde{\rho}_{Gi,j-1}^{\kappa} + \left(\frac{\Delta z_{j-1}}{2\Delta z_{j-1/2}} - \eta_{j-1/2} \right) \tilde{\rho}_{Gi,j}^{\kappa} \right] .$$

In the evaluation of Eqs. (F-12) through (F-19), the pressure derivative of the vapor density is defined only for two-phase cells. Therefore, terms such as

$$\frac{\partial \tilde{\rho}_G^{\kappa}}{\partial p_{i\pm 1,j}} \quad \text{and} \quad \frac{\partial \tilde{\rho}_G^{\kappa}}{\partial p_{i,j\pm 1}}$$

are evaluated only if the respective nodes $[(i\pm 1,j)]$ and $[(i,j\pm 1)]$ are two phase.

For single-phase mesh cells, $\tilde{D}_L$ is the same as the two-phase $\tilde{D}_G$ with the vapor densities and velocities replaced by liquid densities and velocities. Hence, in single-phase cells Eqs. (F-2) and (F-3) become

$$\tilde{D}_L^{\kappa} = \frac{1}{\Delta t} (\tilde{\rho}_L^{\kappa} - \rho_L^n) + \frac{\Delta_i < \tilde{\rho}_L^{\kappa} r_s^{\kappa} \tilde{U}_L^{\kappa} >_j}{r_i^{\kappa} \Delta r_i} + \frac{\Delta_j < \tilde{\rho}_L^{\kappa} \tilde{v}_L^{\kappa} >_i}{\Delta z_j} + \Gamma_L , \text{ and} \quad (F-20)$$

$$\frac{\partial \tilde{D}_{Li,j}^{\kappa}}{\partial p_{ij}} = \frac{1}{\Delta t} \left(\frac{\partial \tilde{\rho}_L^{\kappa}}{\partial p} \right)_{ij} + \frac{1}{r_i^{\frac{1}{2}} \Delta r_i} \left[\frac{\partial < \tilde{\rho}_L^{\kappa} r^{\frac{1}{2}} \tilde{U}_L^{\kappa} >_{i+1/2,j}}{\partial p_{ij}} - \frac{\partial < \tilde{\rho}_L^{\kappa} r^{\frac{1}{2}} \tilde{U}_L^{\kappa} >_{i-1/2,j}}{\partial p_{ij}} \right]$$

(F-21)

$$+ \frac{1}{\Delta z_j} \left[\frac{\partial < \tilde{\rho}_L^{\kappa} \tilde{V}_L^{\kappa} >_{i,j+1/2}}{\partial p_{ij}} - \frac{\partial < \tilde{\rho}_L^{\kappa} \tilde{V}_L^{\kappa} >_{i,j-1/2}}{\partial p_{ij}} \right] .$$

Similar expressions can be written by inspection for Eqs. (F-4) through (F-19), and are not reproduced here. However, it should be noted that in the single-phase analog to Eqs. (F-12) through (F-19), terms such as

$$\frac{\partial \tilde{\rho}_{Li\pm 1,j}^{\kappa}}{\partial p_{i\pm 1,j}} \quad \text{and} \quad \frac{\partial \tilde{\rho}_{Li,j\pm 1}^{\kappa}}{\partial p_{i,j\pm 1}}$$

appear. These are defined only for single-phase cells; hence, they are evaluated only if the respective nodes $(i\pm 1,j)$ and $(i,j\pm 1)$ are single phase.

The remaining undefined quantities in the above expressions are of the derivatives of the velocity components with respect to pressure. The 16 required derivatives are

$$\frac{\partial \tilde{U}_{Gi+1/2,j}^{\kappa}}{\partial p_{ij}}, \quad \frac{\partial \tilde{U}_{Li+1/2,j}^{\kappa}}{\partial p_{ij}}, \quad \frac{\partial \tilde{U}_{Gi-1/2,j}^{\kappa}}{\partial p_{ij}}, \quad \frac{\partial \tilde{U}_{Li-1/2,j}^{\kappa}}{\partial p_{ij}},$$

$$\frac{\partial \tilde{U}_{Gi+1/2,j}^{\kappa}}{\partial p_{i+1,j}}, \quad \frac{\partial \tilde{U}_{Li+1/2,j}^{\kappa}}{\partial p_{i+1,j}}, \quad \frac{\partial \tilde{U}_{Gi-1/2,j}^{\kappa}}{\partial p_{i-1,j}}, \quad \frac{\partial \tilde{U}_{Li-1/2,j}^{\kappa}}{\partial p_{i-1,j}},$$

$$\frac{\partial \tilde{V}_{Gi,j+1/2}^{\kappa}}{\partial p_{ij}}, \quad \frac{\partial \tilde{V}_{Li,j+1/2}^{\kappa}}{\partial p_{ij}}, \quad \frac{\partial \tilde{V}_{Gi,j-1/2}^{\kappa}}{\partial p_{ij}}, \quad \frac{\partial \tilde{V}_{Li,j-1/2}^{\kappa}}{\partial p_{ij}},$$

$$\frac{\partial \tilde{v}_{Gi,j+1/2}^{\kappa}}{\partial p_{i,j+1}}, \quad \frac{\partial \tilde{v}_{Li,j+1/2}^{\kappa}}{\partial p_{i,j+1}}, \quad \frac{\partial \tilde{v}_{Gi,j-1/2}^{\kappa}}{\partial p_{i,j-1}}, \quad \text{and} \quad \frac{\partial \tilde{v}_{Li,j-1/2}^{\kappa}}{\partial p_{i,j-1}}.$$

To determine $\partial \tilde{U}_{Gi+1/2,j}^{\kappa} / \partial p_{ij}$, take the derivative of Eq. (IV-193) with respect to p_{ij} at iteration κ .

$$\begin{aligned} \frac{\partial \tilde{U}_{G\lambda}^{\kappa}}{\partial p_{ij}} = & \frac{1}{D_{\lambda}} [(\tilde{\rho}_{L\lambda}^{\kappa} + \Delta t \{K_{LS\lambda} + K_{GL\lambda} + \Gamma_{LG\lambda} + \Gamma_{LS\lambda}\}) \frac{\partial C_{\lambda}}{\partial p_{ij}} + \frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{ij}} C_{\lambda} \\ & + \Delta t \{K_{GL\lambda} + \Gamma_{LG\lambda}\} \frac{\partial B_{\lambda}}{\partial p_{ij}}] - \frac{\tilde{U}_{G\lambda}^{\kappa}}{D_{\lambda}} \frac{\partial D_{\lambda}}{\partial p_{ij}}, \end{aligned} \quad (F-22)$$

where $\lambda = i+1/2, j$ and D_{λ} is defined by Eq. (IV-201). Only $\tilde{\rho}_{L\lambda}^{\kappa}$, $\tilde{\rho}_{G\lambda}^{\kappa}$, $\tilde{U}_{L\lambda}^{\kappa}$, and $\tilde{U}_{G\lambda}^{\kappa}$ are dependent on p_{ij} within the pressure iteration. The derivatives of Eqs. (IV-197), (IV-199), and (IV-201) with respect to p_{ij} yield

$$\frac{\partial B_{\lambda}}{\partial p_{ij}} = \frac{\Delta t}{\Delta r_{\lambda}} \alpha_{L\lambda}^{n+1},$$

$$\frac{\partial C_{\lambda}}{\partial p_{ij}} = \frac{\Delta t}{\Delta r_{\lambda}} \alpha_{G\lambda}^{n+1}, \quad \text{and}$$

$$\begin{aligned} \frac{\partial D_{\lambda}}{\partial p_{ij}} = & [\tilde{\rho}_{L\lambda}^{\kappa} + \Delta t (K_{LS\lambda} + K_{GL\lambda} + \Gamma_{LG\lambda} + \Gamma_{LS\lambda})] \frac{\partial \tilde{\rho}_{G\lambda}^{\kappa}}{\partial p_{ij}} \\ & + [\tilde{\rho}_{G\lambda}^{\kappa} + \Delta t (K_{GS\lambda} + K_{GL\lambda} + \Gamma_{GL\lambda})] \frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{ij}}. \end{aligned}$$

Substituting these results into Eq. (F-22) and simplifying gives

$$\begin{aligned}
\frac{\partial \tilde{U}_{G\lambda}^{\kappa}}{\partial p_{ij}} = & \frac{1}{D_{\lambda}} \{ [\tilde{\rho}_{L\lambda}^{\kappa} + \Delta t (K_{LS\lambda} + K_{GL\lambda} + \Gamma_{LG\lambda} + \Gamma_{LS\lambda})] [\frac{\Delta t}{\Delta r_{\lambda}} \alpha_{G\lambda}^{n+1} - \tilde{U}_{G\lambda}^{\kappa} \frac{\partial \tilde{\rho}_{G\lambda}^{\kappa}}{\partial p_{ij}}] \\
& + \Delta t (K_{GL\lambda} + \Gamma_{LG\lambda}) \frac{\Delta t}{\Delta r_{\lambda}} \alpha_{L\lambda}^{n+1} - [\tilde{\rho}_{G\lambda}^{\kappa} \tilde{U}_{G\lambda}^{\kappa} - C_{\lambda} + \Delta t (K_{GS\lambda} + K_{GL\lambda} \\
& + \Gamma_{GL\lambda}) \tilde{U}_{G\lambda}^{\kappa}] \frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{ij}} \} .
\end{aligned} \tag{F-23}$$

From Eq. (IV-177), the coefficient of the term

$$\frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{ij}}$$

in Eq. (F-23) can be shown to equal $\Delta t (K_{GL\lambda} + \Gamma_{LG\lambda}) \tilde{U}_{L\lambda}^{\kappa}$. Substituting gives

$$\begin{aligned}
\frac{\partial \tilde{U}_{G\lambda}^{\kappa}}{\partial p_{ij}} = & \frac{1}{D_{\lambda}} \{ [\tilde{\rho}_{L\lambda}^{\kappa} + \Delta t (K_{LS\lambda} + K_{GL\lambda} + \Gamma_{LG\lambda} + \Gamma_{LS\lambda})] [\frac{\Delta t}{\Delta r_{\lambda}} \alpha_{G\lambda}^{n+1} - \tilde{U}_{G\lambda}^{\kappa} \frac{\partial \tilde{\rho}_{G\lambda}^{\kappa}}{\partial p_{ij}}] \\
& + \Delta t (K_{GL\lambda} + \Gamma_{LG\lambda}) (\frac{\Delta t}{\Delta r_{\lambda}} \alpha_{L\lambda}^{n+1} - \tilde{U}_{L\lambda}^{\kappa} \frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{ij}}) \} .
\end{aligned} \tag{F-24}$$

Similar operations applied to $\partial \tilde{U}_{Li+1/2,j}^{\kappa} / \partial p_{ij}$ yield

$$\begin{aligned}
\frac{\partial \tilde{U}_{L\lambda}^{\kappa}}{\partial p_{ij}} = & \frac{1}{D_{\lambda}} \{ [\tilde{\rho}_{G\lambda}^{\kappa} + \Delta t (K_{GS\lambda} + K_{GL\lambda} + \Gamma_{GL\lambda})] [\frac{\Delta t}{\Delta r_{\lambda}} \alpha_{L\lambda}^{n+1} - \tilde{U}_{L\lambda}^{\kappa} \frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{ij}}] \\
& + \Delta t (K_{GL\lambda} + \Gamma_{GL\lambda}) (\frac{\Delta t}{\Delta r_{\lambda}} \alpha_{G\lambda}^{n+1} - \tilde{U}_{G\lambda}^{\kappa} \frac{\partial \tilde{\rho}_{G\lambda}^{\kappa}}{\partial p_{ij}}) \} .
\end{aligned} \tag{F-25}$$

Based on the definitions of $\tilde{\rho}_{L\lambda}$ and $\tilde{\rho}_{G\lambda}$ in Eq. (IV-191) and (IV-189),

$$\frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{ij}} = \begin{cases} \frac{1}{2} \frac{\Delta r_{i+1}}{\Delta r_{\lambda}} \frac{\partial \tilde{\rho}_{Lij}^{\kappa}}{\partial p_{ij}} = \frac{1}{2} \frac{\Delta r_{i+1}}{\Delta r_{\lambda}} \alpha_{Lij}^{n+1} \frac{\partial \rho_{Lij}}{\partial p_{ij}} & , \text{ for single-phase mesh cell } i,j \\ 0 & , \text{ for two-phase mesh cell } i,j \end{cases}$$

$$\frac{\partial \tilde{\rho}_{G\lambda}^{\kappa}}{\partial p_{ij}} = \begin{cases} 0 & , \text{ for single-phase mesh cell } i,j \\ \frac{1}{2} \frac{\Delta r_{i+1}}{\Delta r_{\lambda}} \frac{\partial \tilde{\rho}_{Gij}^{\kappa}}{\partial p_{ij}} = \frac{1}{2} \frac{\Delta r_{i+1}}{\Delta r_{\lambda}} \alpha_{Gij}^{n+1} \frac{\partial \rho_{Gij}}{\partial p_{ij}} & , \text{ for two-phase mesh cell } i,j. \end{cases}$$

The derivative of these λ interface radial velocity components with respect to $p_{i+1,j}$ has a form similar to Eqs. (F-24) and (F-25) except that the pressure gradient term derivatives

$$\frac{2B_{\lambda}}{2p_{i+1,j}} \quad \text{and} \quad \frac{2C_{\lambda}}{2p_{i+1,j}}$$

are of opposite sign, and the $\tilde{\rho}_{G\lambda}^{\kappa}$ and $\tilde{\rho}_{L\lambda}^{\kappa}$ derivatives use the parts of $\tilde{\rho}_{G\lambda}^{\kappa}$ and $\tilde{\rho}_{L\lambda}^{\kappa}$ that derive from mesh cell $i+1,j$. Therefore, these expressions are

$$\begin{aligned} \frac{\partial \tilde{u}_{G\lambda}^{\kappa}}{\partial p_{i+1,j}} = \frac{1}{D_{\lambda}} \{ [\tilde{\rho}_{L\lambda} + \Delta t (K_{LS\lambda} + K_{GL\lambda} + \Gamma_{LG\lambda} + \Gamma_{LS\lambda})] [-\frac{\Delta t}{\Delta r_{\lambda}} \alpha_{G\lambda}^{n+1} - \tilde{u}_{G\lambda}^{\kappa} \frac{\partial \tilde{\rho}_{G\lambda}^{\kappa}}{\partial p_{i+1,j}}] \\ + \Delta t (K_{GL\lambda} + \Gamma_{LG\lambda}) (-\frac{\Delta t}{\Delta r_{\lambda}} \alpha_{L\lambda}^{n+1} - \tilde{u}_{L\lambda}^{\kappa} \frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{i+1,j}}) \} \quad , \text{ and} \quad (F-26) \end{aligned}$$

$$\frac{\partial \tilde{u}_{L\lambda}^{\kappa}}{\partial p_{i+1,j}} = \frac{1}{D_{\lambda}} \{ [\tilde{\rho}_{G\lambda} + \Delta t (K_{GS\lambda} + K_{GL\lambda} + \Gamma_{GL\lambda} + \Gamma_{LS\lambda})] [-\frac{\Delta t}{\Delta r_{\lambda}} \alpha_{L\lambda}^{n+1} - \tilde{u}_{L\lambda}^{\kappa} \frac{\partial \tilde{\rho}_{L\lambda}^{\kappa}}{\partial p_{i+1,j}}] \}$$

$$+ \Delta t (K_{GL\lambda} + \Gamma_{GL\lambda}) \left(-\frac{\Delta t}{\Delta r_\lambda} \alpha_{G\lambda}^{n+1} - \tilde{U}_{G\lambda}^\kappa \frac{\partial \tilde{\rho}_{G\lambda}^\kappa}{\partial p_{i+1,j}} \right) \quad , \quad (F-27)$$

where

$$\frac{\partial \tilde{\rho}_{L\lambda}^\kappa}{\partial p_{i+1,j}} = \begin{cases} \frac{1}{2} \frac{\Delta r_i}{\Delta r_\lambda} \frac{\partial \tilde{\rho}_{Li+1,j}^\kappa}{\partial p_{i+1,j}} = \frac{1}{2} \frac{\Delta r_i}{\Delta r_\lambda} \alpha_{Li+1,j}^{n+1} \frac{\partial \rho_{Li+1,j}}{\partial p_{i+1,j}} & , \text{ for single-phase mesh cell } i+1,j \\ 0 & , \text{ for two-phase mesh cell } i+1,j \end{cases}$$

and

$$\frac{\partial \tilde{\rho}_{G\lambda}^\kappa}{\partial p_{i+1,j}} = \begin{cases} 0 & , \text{ for single-phase mesh cell } i+1,j \\ \frac{1}{2} \frac{\Delta r_i}{\Delta r_\lambda} \frac{\partial \tilde{\rho}_{Gi+1,j}^\kappa}{\partial p_{i+1,j}} = \frac{1}{2} \frac{\Delta r_i}{\Delta r_\lambda} \alpha_{Gi+1,j}^{n+1} \frac{\partial \rho_{Gi+1,j}}{\partial p_{i+1,j}} & , \text{ for two-phase mesh cell } i+1,j. \end{cases}$$

The expressions for

$$\frac{\partial \tilde{U}_{Gi-1/2,j}^\kappa}{\partial p_{i-1,j}} \quad , \quad \frac{\partial \tilde{U}_{Li-1/2,j}^\kappa}{\partial p_{i-1,j}} \quad , \quad \frac{\partial \tilde{U}_{Gi-1/2,j}^\kappa}{\partial p_{ij}} \quad , \quad \text{and} \quad \frac{\partial \tilde{U}_{Li-1/2,j}^\kappa}{\partial p_{ij}}$$

are the same as Eqs. (F-24) through (F-27), respectively, but with i replaced by $i-1$, λ replaced by $i-1/2$, and $i+1$ replaced by i . The remaining derivatives of the axial velocity components for pressure

$$\frac{\partial \tilde{V}_{G\theta}^\kappa}{\partial p_{ij}} \quad , \quad \frac{\partial \tilde{V}_{L\theta}^\kappa}{\partial p_{ij}} \quad , \quad \frac{\partial \tilde{V}_{G\theta}^\kappa}{\partial p_{i,j+1}} \quad , \quad \frac{\partial \tilde{V}_{L\theta}^\kappa}{\partial p_{i,j+1}} \quad ,$$

$$\frac{\partial \tilde{V}_{Gi,j-1/2}^\kappa}{\partial p_{i,j-1}} \quad , \quad \frac{\partial \tilde{V}_{Li,j-1/2}^\kappa}{\partial p_{i,j-1}} \quad , \quad \frac{\partial \tilde{V}_{Gi,j-1/2}^\kappa}{\partial p_{ij}} \quad , \quad \text{and} \quad \frac{\partial \tilde{V}_{Li,j-1/2}^\kappa}{\partial p_{ij}} \quad ,$$

where $\theta = i, j+1/2$, also are defined by expressions similar to Eqs. (F-24) through (F-27). They can be obtained by simply replacing U by V , Δr by Δz , λ by θ , $i+1, j$ by $i, j+1$, $i-1/2, j$ by $i, j-1/2$, and $i-1, j$ by $i, j-1$. One further modification must be made to account for the gravity acceleration term acting in the axial direction. The derivatives of B_θ and C_θ for pressure have the modified form

$$\frac{\partial B_\theta}{\partial p_{ij}} = \frac{\Delta t}{\Delta z_\theta} \alpha_{L\theta}^{n+1} + \Delta t \, g \, \frac{\partial \tilde{\rho}_{L\theta}}{\partial p_{ij}} ,$$

$$\frac{\partial C_\theta}{\partial p_{ij}} = \frac{\Delta t}{\Delta z_\theta} \alpha_{G\theta}^{n+1} + \Delta t \, g \, \frac{\partial \tilde{\rho}_{G\theta}}{\partial p_{ij}} ,$$

$$\frac{\partial B_\theta}{\partial p_{i,j+1}} = -\frac{\Delta t}{\Delta z_\theta} \alpha_{L\theta}^{n+1} + \Delta t \, g \, \frac{\partial \tilde{\rho}_{L\theta}}{\partial p_{i,j+1}} , \text{ and}$$

$$\frac{\partial C_\theta}{\partial p_{i,j+1}} = -\frac{\Delta t}{\Delta z_\theta} \alpha_{G\theta}^{n+1} + \Delta t \, g \, \frac{\partial \tilde{\rho}_{G\theta}}{\partial p_{i,j+1}} .$$

Thus, in the derivation of

$$\frac{\partial \tilde{v}_{G\theta}^\kappa}{\partial p_{ij}} , \frac{\partial \tilde{v}_{L\theta}^\kappa}{\partial p_{ij}} , \frac{\partial \tilde{v}_{G\theta}^\kappa}{\partial p_{i,j+1}} , \text{ and } \frac{\partial \tilde{v}_{L\theta}^\kappa}{\partial p_{i,j+1}} ,$$

the $-\tilde{v}_{G\theta}^\kappa$ coefficient of

$$\frac{\partial \tilde{p}_{G\theta}^\kappa}{\partial p_{ij}} \text{ and } \frac{\partial \tilde{p}_{G\theta}^\kappa}{\partial p_{i,j+1}}$$

and the $-\tilde{V}_{L\theta}^{\kappa}$ coefficient of

$$\frac{\partial \tilde{\rho}_{L\theta}^{\kappa}}{\partial p_{ij}} \text{ and } \frac{\partial \tilde{\rho}_{L\theta}^{\kappa}}{\partial p_{i,j+1}}$$

are modified to $(\Delta t g - \tilde{V}_{G\theta}^{\kappa})$ and $(\Delta t g - \tilde{V}_{L\theta}^{\kappa})$, respectively.

This completes the definition of the terms needed to evaluate the IBAR times JBAR set of equations for all mesh cells in the problem. These equations are solved by the matrix solution techniques described in Appendix G. The boundary cells are evaluated by the following conditions.

1. On the left boundary, $\delta \tilde{p}_{i-1,j} = \delta \tilde{p}_{i,j}$.
2. On the bottom boundary, $\delta \tilde{p}_{i,j-1} = 0$ if the pressure is given as a function of time. Otherwise, $\delta \tilde{p}_{i,j-1} = \delta \tilde{p}_{i,j}$.
3. On the right boundary, $\delta \tilde{p}_{i+1,j} = 0$ if the pressure is given as a function of time. Otherwise, $\delta \tilde{p}_{i+1,j} = \delta \tilde{p}_{i,j}$.
4. On the top boundary, $\delta \tilde{p}_{i,j+1} = 0$ if the pressure is given as a function of time. Otherwise, $\delta \tilde{p}_{i,j+1} = \delta \tilde{p}_{i,j}$.

Finally, if a rigid can wall exists on a mesh cell radial interface, its radial velocities are zero by definition. All matrix coefficients and terms involving radial effects of pressure are zero in this case.

APPENDIX G

MATRIX INVERSION METHODS

The pressure-change and liquid- and vapor-field component density implicit matrix equations involve five-point, two-dimensional spatial coupling that requires a simultaneous matrix solution. An efficient matrix inversion method is needed to accomplish this. Two different methods have been programmed in SIMMER-II: Successive Line Overrelaxation (SLOR)^{1,2} and an algorithm called the Approximate Inversion Method (AIM).³ To generalize these solution procedures for the following discussion, the pressure-change matrix equation

$$\underline{\underline{B}}^k \cdot \underline{\underline{\delta p}}^k = -\underline{\underline{\tilde{D}}}^k \quad (G-1)$$

and the liquid- and vapor-field component density matrix equations

$$\begin{aligned} \underline{\underline{A}}_F^{n+1} \cdot \underline{\underline{p}}_{Fm}^{n+1} &= (\underline{\underline{p}}_{Fm}^n - \Delta t \underline{\underline{r}}_{Fm}^n) , & F = \ell, m = 1, \dots, NCLR \\ & & F = g, m = 1, \dots, NCGR \end{aligned} \quad (G-2)$$

are recast into the following general form:

$$\underline{\underline{M}} \cdot \underline{\underline{F}} = \underline{\underline{Q}} \quad (G-3)$$

Matrix $\underline{\underline{M}}$ and vector $\underline{\underline{Q}}$ are known; vector $\underline{\underline{F}}$ is the unknown to be determined.

SLOR is a popular method for solving this matrix equation because it is simple and efficient. The method is documented extensively in the literature,^{1,2} so we will not describe it in detail. However, for completeness, a summary of the working equations and a discussion of unique features in their application is given. The method is based on partitioning the block tridiagonal matrix $\underline{\underline{M}}$ into the sum of three matrices representing the block diagonals, $\underline{\underline{M}} = \underline{\underline{L}} + \underline{\underline{D}} + \underline{\underline{U}}$. Matrix $\underline{\underline{D}}$ has tridiagonal element submatrix blocks accounting for the spatial coupling of a line of mesh cells. Matrices $\underline{\underline{L}}$ and $\underline{\underline{U}}$ have diagonal element submatrix blocks accounting for spatial coupling to the neighbor mesh cells in the previous line and next line, respectively. Rearranging the matrix

equation as an iterative equation and extrapolating the estimate of the solution at each iteration k with a relaxation parameter ω gives the SLOR iterative algorithm as

$$\underline{\hat{F}}^k = \underline{D}^{-1} \cdot (\underline{Q} - \underline{L} \cdot \underline{\hat{F}}^k - \underline{U} \cdot \underline{F}^{k-1}) \quad , \quad (G-4)$$

followed by

$$\underline{F}^k = \underline{F}^{k-1} + \omega(\underline{\hat{F}}^k - \underline{F}^{k-1}) \quad . \quad (G-5)$$

Starting with an initial estimate of the solution, $\underline{F}^0$, the iterative evaluation continues until the convergence criterion,

$$\frac{(\underline{F}^k - \underline{F}^{k-1}, \underline{F}^k - \underline{F}^{k-1})}{(\underline{F}^k, \underline{F}^k)} < \epsilon \quad , \quad (G-6)$$

is satisfied for some small number ϵ . The user also defines the number of iterations between updating the relaxation parameter ω based on^{1,2}

$$\omega = \frac{2}{1 + (1 - \mu_0^2)^{1/2}} \quad , \quad (G-7)$$

where

$$\mu_0 = \frac{\mu + \omega_{\text{previous}}^{-1}}{\omega_{\text{previous}} (\mu)^{1/2}} \quad \text{and}$$

μ = the iteration matrix spectral radius, or

$$\mu \cong \left\{ \frac{(\underline{F}^k - \underline{F}^{k-1}, \underline{F}^k - \underline{F}^{k-1})}{(\underline{F}^{k-1} - \underline{F}^{k-2}, \underline{F}^{k-1} - \underline{F}^{k-2})} \right\}^{1/2} \quad .$$

The initial value of ω is set to unity. Infrequently, the updated value of ω can cause the iterative solution to diverge. This occurs when the Euclidean norm error ratio overestimates μ , causing ω to be too large. This condition is corrected by setting ω to unity for the next update cycle. At the end of the cycle, the maximum of the current Euclidean norm error ratio and 90% of its value before divergence occurred are used to define the next ω .

The AIM is a second method that can be used to solve the matrix equation. It is based on modifying $\underline{\underline{M}}$ to achieve an efficient direct inversion. Error in the solution from modifying $\underline{\underline{M}}$ is removed iteratively. The existing error at iteration k is

$$\underline{\underline{\Delta Q}}^k = \underline{\underline{Q}} - \underline{\underline{M}} \cdot \underline{\underline{F}}^k . \quad (\text{G-8})$$

The change in $\underline{\underline{F}}^k$ needed to remove this error is defined by the matrix equation

$$\underline{\underline{M}} \cdot \underline{\underline{\Delta F}}^k = \underline{\underline{\Delta Q}}^k . \quad (\text{G-9})$$

However, modifying $\underline{\underline{M}}$ during its inversion gives only an approximate solution:

$$\underline{\underline{\Delta F}}^k \cong (\underline{\underline{M}}^{\text{approx}})^{-1} \cdot \underline{\underline{\Delta Q}}^k . \quad (\text{G-10})$$

Because $\underline{\underline{\Delta F}}^k$ is approximate, combining it with $\underline{\underline{F}}^k$ gives only the next iteration value for $\underline{\underline{F}}$, that is, $\underline{\underline{F}}^{k+1}$. To accelerate convergence of $\underline{\underline{F}}^{k+1}$ to $\underline{\underline{F}}$, a diagonal-element relaxation matrix $\underline{\underline{W}}$ is applied to $\underline{\underline{\Delta F}}^k$:

$$\underline{\underline{F}}^{k+1} = \underline{\underline{F}}^k + \underline{\underline{W}} \cdot \underline{\underline{\Delta F}}^k . \quad (\text{G-11})$$

Three definitions of $\underline{\underline{W}}$ are used in SIMMER-II:

1. $\omega_{\ell\ell} = 1$, no relaxation;
2. $\omega_{\ell\ell} = \omega$, global rebalance relaxation; and

3. $\omega_{\ell\ell} = \omega_j$, line rebalance relaxation, where $j = 1 + [(\ell-1)/I]$, I is the number of mesh cells in a line, and j is truncated to an integer value.

No relaxation is applied the first NITSRU iterations; overall rebalance relaxation is applied the next NITSRU iterations, and line rebalance relaxation is applied thereafter. NITSRU is specified by the user. The ω and ω_j rebalance parameters are computed to minimize the remaining Euclidean norm error in $\underline{Q}$,

$$\text{Error} = (\underline{\Delta Q}^{k+1}, \underline{\Delta Q}^{k+1})$$

$$= \sum_{\ell}^{I*J} (\underline{\Delta Q}^{k+1})^2 \quad (\text{G-12})$$

$$= \sum_{\ell}^{I*J} (\underline{\Delta Q}_{\ell}^k - \omega_{\ell\ell} \underline{\Delta Q}_{\ell\text{est}}^k)^2 \quad \text{where } \underline{\Delta Q}_{\text{est}}^k = \underline{M} \cdot \underline{\Delta F}^k .$$

The error is minimized with respect to ω or ω_j by requiring $d\text{Error}/d\omega = 0$ or $\partial\text{Error}/\partial\omega_j = 0$ for $j = 1, 2, \dots, J$, respectively. The $d\text{Error}/d\omega = 0$ requirement yields

$$\omega = \frac{\sum_{\ell}^{I*J} \underline{\Delta Q}_{\ell}^k \underline{\Delta Q}_{\ell\text{est}}^k}{\sum_{\ell}^{I*J} (\underline{\Delta Q}_{\ell\text{est}}^k)^2} . \quad (\text{G-13})$$

The $\partial\text{Error}/\partial\omega_j = 0$ requirement of J equations gives a five-diagonal element coefficient matrix equation with spatial coupling to the four nearest mesh-cell lines. This matrix equation is solved for ω_j , $j = 1, 2, \dots, J$ by forward elimination of the two lower diagonals and backward substitution into the two modified upper diagonals. The iteration continues with relaxation of $\underline{F}^{k+1}$ by one of the methods for determining $\underline{W}$ until the convergence criterion,

$$\frac{(\underline{\Delta Q}^i, \underline{\Delta Q}^i)}{(\underline{Q}, \underline{Q})} < \epsilon , \quad (\text{G-14})$$

is satisfied for the input precision criterion ϵ .

The AIM algorithm for modifying and inverting $\underline{\underline{M}}$ is the familiar Crout-Cholesky forward-elimination and backward-substitution method.⁴ Rather than operating on individual elements, AIM operates on submatrix blocks. The block tridiagonal matrix $\underline{\underline{M}} = \underline{\underline{L}} + \underline{\underline{D}} + \underline{\underline{U}}$ is inverted by forward-eliminating its lower block diagonal $\underline{\underline{L}}$ and backward-substituting into its modified upper block diagonal $\underline{\underline{U}}^*$. This procedure is made efficient by a submatrix collapsing approximation.

To picture this in more detail, first consider the forward-elimination of $\underline{\underline{L}}$. The procedure involves stepping along the $j = 1, 2, \dots, J-1$ mesh cell lines, eliminating the $(j+1)^{\text{th}}$ mesh cell line submatrix $\underline{\underline{L}}_{j+1}$. Start by defining the submatrix and vector

$$\underline{\underline{D}}_1^* = \underline{\underline{D}}_1, \text{ and} \quad (\text{G-15})$$

$$\underline{\underline{\Delta Q}}_1^{*k} = \underline{\underline{\Delta Q}}_1^k \quad (\text{G-16})$$

for the first mesh cell line. Then for each j^{th} mesh cell line, starting with $j = 1$, the tridiagonal element submatrix $\underline{\underline{D}}_j^*$ is inverted directly by the Crout-Cholesky forward-elimination and backward-substitution method. The remaining elements of the j mesh cell line then are multiplied by $(\underline{\underline{D}}_j^*)^{-1}$. The $\underline{\underline{D}}_j^*$ matrix becomes the identity matrix; $\underline{\underline{U}}_j$ becomes $(\underline{\underline{D}}_j^*)^{-1} \cdot \underline{\underline{U}}_j$, a full element submatrix; and $\underline{\underline{\Delta Q}}_j^{*k}$ becomes

$$\underline{\underline{\Delta Q}}_j^{*k} = (\underline{\underline{D}}_j^*)^{-1} \cdot \underline{\underline{\Delta Q}}_j^{*k} . \quad (\text{G-17})$$

An element-collapsing approximation now is made on the full submatrix $(\underline{\underline{D}}_j^*)^{-1} \cdot \underline{\underline{U}}_j$. Row elements to the left and right of the submatrix diagonal element are collapsed (summed) to the adjacent left and right elements, respectively:

$$(\underline{\underline{D}}_j^*)^{-1} \cdot \underline{\underline{U}}_j \xrightarrow[\text{approximation}]{\text{collapse}} \underline{\underline{U}}_j^* . \quad (\text{G-18})$$

Consequently, $(\underline{D}_j^*)^{-1} \cdot \underline{U}_j$ becomes the tridiagonal element submatrix $\underline{U}_j^*$. $\underline{L}_{j+1}$ now can be eliminated from the $(j+1)^{\text{th}}$ row submatrix equation by multiplying the modified j^{th} row submatrix equation by the diagonal element submatrix $\underline{L}_{j+1}$ and then subtracting it from the $(j+1)^{\text{th}}$ row equation, giving

$$\underline{D}_{j+1}^* = \underline{D}_{j+1} - \underline{L}_{j+1} \cdot \underline{U}_j^* \quad \text{and} \quad (\text{G-19})$$

$$\underline{\Delta Q}_{j+1}^{+k} = \underline{\Delta Q}_{j+1}^k - \underline{L}_{j+1} \underline{\Delta Q}_j^{*k} \quad (\text{G-20})$$

At this point, the reason for collapsing $(\underline{D}_j^*)^{-1} \cdot \underline{U}_j$ to the tridiagonal element submatrix $\underline{U}_j^*$ becomes evident. The tridiagonal element form of $\underline{D}_{j+1}$ is maintained during its modification to $\underline{D}_{j+1}^*$. Without the collapsing approximation, $\underline{D}_{j+1}^*$ would be a full element submatrix. Inverting $\underline{D}_{j+1}^*$ in the next step now can be done efficiently by the Crout-Cholesky method rather than by a full element submatrix inversion.

After stepping forward with the above procedure, in other words, inverting $\underline{D}_j^*$ and then applying Eqs. (G-17) to (G-20) for each $j = 1, 2, \dots, J-1$ line, the solution estimate in the J^{th} (last) mesh cell line is determined by

$$\underline{\Delta F}_J^k = (\underline{D}_J^*)^{-1} \underline{\Delta Q}_J^{+k} \quad (\text{G-21})$$

At this point, $\underline{L}$ has been eliminated, $\underline{D}$ has become the identity matrix $\underline{I}$, and $\underline{U}$ has been modified to $\underline{U}^*$ having tridiagonal element submatrix blocks. To complete the solution, backwards-substitution into the $\underline{I} + \underline{U}^*$ upper triangular matrix is used to determine

$$\underline{\Delta F}_j^k = \underline{\Delta Q}_j^{*k} - \underline{U}_j^* \underline{\Delta F}_{j+1}^k \quad (\text{G-22})$$

for $j = J-1, J-2, \dots, 1$. $\underline{F}^{k+1}$ can now be computed from Eq. (G-11) using one of the three indicated methods for determining $\underline{W}$.

Applying AIM with no relaxation on a strongly coupled (weakly diagonally dominant) matrix equation can result in a divergent algorithm. This occurs when the approximation to the inverse of $\underline{M}$ loses diagonal dominance. Divergence can be prevented in two ways. The approximate inverse for $\underline{M}$ can be constrained to

maintain diagonal dominance through further modification. The other way, which is applied in SIMMER-II, is to use rebalance relaxation when no relaxation leads to divergence. A small value (5 to 10) for NITSRU implements this feature automatically. The stabilizing effect of rebalance relaxation has been found to override any instability caused by inverting $\underline{M}$.

In implementing the SLOR and AIM algorithms in SIMMER-II, two common features have been introduced. To improve the algorithms' efficiency for Eq. (G-2), the coefficient matrix is inverted only for the first component in a velocity field. In treating subsequent components, the inverted matrix is applied to the right-side vector, which reduces the calculative effort per iteration by 40%. For one-dimensional problems, both methods use the same Crout-Choleksy direct matrix inversion algorithm.

The MATSOL option to the preprocessor implements either method in SIMMER-II. With MATSOL on, SLOR is used; with MATSOL off, AIM is used. AIM requires twice the variable storage that SLOR requires. When LCM is used to store fluid-state parameters, no penalty for extra AIM storage is involved.

A numerical study was performed to compare the efficiency of SLOR and AIM as applied in SIMMER-II. Transition-phase and work-energy test problems were used for this comparison. Can-wall restraint on radial motion was removed in both problems to have two-dimensional spatial coupling.

The number of iterations and calculation times of AIM and SLOR were monitored when evaluating the two test problems. SLOR required four times as many iterations and twice the calculative effort of AIM with no relaxation to solve the pressure change matrix equation. Equal computation times were required for the material density matrix equations because SLOR required only two iterations to converge compared with one iteration for AIM. Applying global and line rebalance relaxation in AIM increased calculative effort per iteration by 20% and 30%, respectively. For these small test problems, AIM with no rebalance relaxation was more efficient. However, for large problems with strong spatial coupling, the rebalance options become more efficient. This occurs when their ability to enhance convergence and reduce the number of iterations overcomes their additional calculative effort per iteration. An example of how effective rebalance relaxation can be was obtained from a realistic 16-by-31 mesh-cell transition-phase analysis problem.⁵ Liquid fuel was predicted to enter the coolant channel and cause large local pressures. The SIMMER-II run aborted when the pressure-change solution evaluated by AIM with no relaxation diverged. This was caused by the loss of diagonal dominance through modifying $\underline{M}$.

during its inversion. Restarting SIMMER-II using the SLOR method required several hundred iterations to obtain a converged solution. However, restarting SIMMER-II using the AIM method with global rebalance relaxation required fewer than 10 iterations to converge the pressure change solution.

REFERENCES

1. L. A. Hageman, "The Estimation of Acceleration Parameters for the Chebyshev Polynomial and the Successive Overrelaxation Iteration Methods," Bettis Atomic Power Laboratory report WAPD-TM-1038 (June 1972).
2. R. S. Varga, Matrix Iterative Analysis (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1962), p. 57.
3. R. G. Steinke, "An Approximate Inversion Method for Five Point Difference Matrix Equations," in Proc. of ANS Topical Meeting on Computational Methods in Nuclear Engineering, Vol. 3 pp. 10-1, 10-11 (April 1979).
4. P. J. Roache, Computational Fluid Dynamics (Hermosa Publishers, Albuquerque, New Mexico, 1976 Rev.), pp. 215-221.
5. J. F. Jackson and M. G. Stevenson, "Nuclear Reactor Safety Quarterly Progress Report, April 1 - June 30, 1978," Los Alamos Scientific Laboratory report LA-7481-PR, NUREG/CR-385 (October 1978).

APPENDIX H

TRANSIENT INITIAL STATE CALCULATION FOR THE SPATIAL NEUTRONICS

The initiating phase of an accident usually is calculated with computer codes that use a neutronics treatment based on point kinetics. SIMMER-II calculations of later accident phases involving significant cladding and fuel motion require initial conditions describing the transient reactor state. Initialization of the SIMMER-II fluid dynamics models for a transient state formally presents few problems because only instantaneous values and non-path-dependent integrals are involved. However, initialization of the spatial neutronics calculations involves the path-dependent delayed-neutron precursor concentrations. In addition, point-kinetics treatments normally are used for the initiating phase, where relatively small neutronic perturbations are expected to occur. These point-kinetics results depend on the values of neutronics parameters, for example, reactivity and delayed neutron fraction, that describe the reactor state. However, inconsistencies may arise because the SIMMER-II spatial neutronics model may not compute identical values because of transient changes or data differences. Thus, a special treatment is necessary for the transient state initialization of the SIMMER-II neutronics algorithm.

The quasi-static approach is used as a basis for the method because this approach provides relations between the integral quantities used in the point kinetics equation and the results of space-dependent calculations. Factors are computed to adjust the basic physical neutronics data, permitting continuity of the solution between the initiating-phase calculation and the SIMMER calculation. This treatment is analogous to the eigenvalue adjustment of the fission neutron yield to achieve an initial stationary solution.

The normal development of the quasi-static method yields the following relations for the integral quantities used in the point kinetics, or more precisely, the amplitude equations. The reactivity is defined as

$$\rho = \frac{1}{\beta} \langle \psi_0^*, [-M + \frac{1}{k_0} (F_p + \sum_d \lambda_d F_d)] \psi \rangle, \quad (H-1)$$

where ψ_0^* indicates the initial stationary-state adjoint flux, k_0 is in the initial stationary-state eigenvalue, and ψ is the time-dependent real flux shape. Equation (H-1) is equivalent to Eq. (V-38) except that it is written in

operator notation, where M denotes the neutron leakage, absorption, and scattering operator; F_p denotes the prompt fission source operator; and F_d denotes the delayed-neutron source operator.

The weighted total fission source, F , is given by

$$F = \frac{1}{k_0} \langle \psi_0^*, (F_p + \sum_d x_d F_d) \psi \rangle .$$

The effective delayed-neutron fractions are defined as

$$\beta_k = \frac{1}{F} \langle \psi_0^*, \frac{1}{k_0} (x_d F_d) \psi \rangle . \quad (H-2)$$

The effective neutron generation time is given as

$$\Lambda = \frac{1}{F} \langle \psi_0^*, \frac{1}{V} \psi \rangle . \quad (H-3)$$

The effective delayed-neutron precursor concentrations are given as

$$\eta_k = \langle \psi_0^*, x_d C_d \rangle . \quad (H-4)$$

The spatial flux shape and delayed precursor concentrations used in Eqs. (H-1) to (H-4) are obtained by solving

$$\frac{1}{V} \frac{\partial \psi}{\partial t} = \left(-M + \frac{1}{k_0} F_p - \frac{\dot{N}}{N} \frac{1}{V} \right) \psi + \frac{1}{N} \sum_d \lambda_d x_d C_d \quad (H-5)$$

and

$$\frac{\partial}{\partial t} C_d = \frac{1}{k_0} F_d \psi - \lambda_d C_d \quad , \quad (H-6)$$

respectively. The values for the quantities on the left side of Eqs. (H-1) to (H-4) are known from the results of the initiating-phase transient solution. To obtain agreement between these prescribed quantities and the SIMMER-II calculated quantities, data adjustment factors are defined for the physical delayed-neutron fraction and the energy group velocities. The stationary-state eigenvalue, k_0 , also must be determined.

The reactivity, ρ_t^i , of the transient system at the time of initialization is related to the initial state of the stationary system through the eigenvalue, k_0 . By rearranging Eq. (H-1), the stationary-state eigenvalue is

$$k_0 = \frac{(1 - \rho_t^i) \langle \psi^*, (F_p + \sum_d h_d \lambda_d F_d) \psi \rangle}{\langle \psi^*, M \psi \rangle} \quad . \quad (H-7)$$

The factor h_d in Eq. (H-7) is used to adjust the delayed-neutron yields and is defined in the next paragraph. The stationary-state adjoint flux, ψ_0^* , in Eqs. (H-1) to (H-4) is not available and is replaced by a stationary adjoint solution, ψ^* , of the transient system at the start of the SIMMER-II calculation. If only minor perturbations* have occurred to the system, ψ^* and ψ_0^* will have nearly the same shape. Although use of Eq. (H-7) in the outer iterations for the initial flux shape can yield rapid convergence, the computed value of k_0 is sensitive to numerical roundoff errors and possible divergence. Therefore, the power iteration [Eq. (V-85)] is used to calculate the initial state eigenvalue.

Equation (H-2) is modified to permit agreement between the calculated and prescribed effective delayed-neutron fractions:

*Actually, this condition is not formally necessary because the adjoint flux is used only as a weighting function. However, consistency with the conventional definition of the reactivity requires the stationary-state adjoint flux as a weighting function.

$$\beta_{t,d}^i = \frac{\langle \psi^*, \frac{1}{k_0} (h_d x_d F_d) \rangle}{\langle \psi^*, \frac{1}{k_0} (F_p + \sum_d h_d x_d F_d) \psi \rangle}, \quad (\text{H-8})$$

where h_d adjusts the physical delayed-neutron yield. By rearranging Eq. (H-8), the factor h_d is defined implicitly as

$$h_d = \beta_{t,d}^i \frac{\langle \psi^*, (F_p + \sum_d h_d x_d F_d) \psi \rangle}{\langle \psi^*, x_d F_d \psi \rangle}. \quad (\text{H-9})$$

Agreement between the calculated and prescribed neutron generation time is achieved by modifying Eq. (H-3) to

$$\Lambda = \frac{\langle \psi^*, \frac{f}{V} \psi \rangle}{\langle \psi^*, \frac{1}{k_0} (F_p + \sum_d h_d x_d F_d) \psi \rangle}, \quad (\text{H-10})$$

where f adjusts the group velocities. By rearranging Eq. (H-10), the factor f is given by

$$f = \Lambda_t^i \frac{\langle \psi^*, \frac{1}{k_0} (F_p + \sum_d h_d x_d F_d) \psi \rangle}{\langle \psi^*, \frac{1}{V} \psi \rangle}. \quad (\text{H-11})$$

The spatial flux shape and precursor shapes are obtained by solving Eqs. (H-5) and (H-6), respectively. Both the integrated precursor densities, $\eta_{t,d}^i$, and the amplitude derivative, $\dot{N}^i$, are known at the initialization time, but the precursor spatial shape and the shape function derivative are unknown. These difficulties are overcome by neglecting the shape function derivative, that is, by assuming that

$$\frac{\partial \psi}{\partial t} = 0 \quad , \quad (H-12)$$

and by assuming that the spatial shape of the precursors is proportional to the delayed-neutron fission source shape; that is,

$$\frac{1}{N} \lambda_d C_d \propto x_d F_d \psi \quad . \quad (H-13)$$

The shape function derivative is assumed to be small during the initiating phase transient. Similarly, the spatial precursor shape is approximated well by the spatial shape of the delayed-neutron source.

The proportionality factor, g_d [implied by Eq. (H-13)], is obtained by integration; that is,

$$g_d = \frac{\langle \psi^*, \lambda_d C_d \rangle \frac{1}{N}}{\langle \psi^*, \frac{1}{k_0} x_d h_d F_d \psi \rangle} \quad . \quad (H-14)$$

By using Eq. (H-4), Eq. (H-14) becomes

$$g_d = \frac{\lambda_d (\eta_{t,d}^i / N^i)}{\langle \psi^*, \frac{1}{k_0} h_d x_d F_d \psi \rangle} \quad . \quad (H-15)$$

Combining Eqs. (H-5), (H-12), and (H-14) yields a modified flux-shape equation:

$$0 = \left(-M + \frac{1}{k_0} F_p - \frac{\dot{N}}{N} \frac{1}{V} \right) \psi + \sum_d g_d \frac{1}{k_0} x_d F_d \psi \quad . \quad (H-16)$$

Although Eq. (H-16) is an inhomogeneous equation, it is independent of amplitude; therefore, ψ can be scaled so that N is equal to unity.

Equations (V-85), (H-8), (H-9), (H-11), (H-15), and (H-16) are solved iteratively for the initial flux shape, ψ ; stationary-state eigenvalue, k_0 ; the data adjustment factors, h_d and f ; and the delayed-neutron source normalization factor, g_k . The normal fission source, or "outer," iteration is used. Experience has shown that the nonlinearities involved do not affect the convergence properties of either the standard power iteration or the Chebyshev accelerated iteration significantly.

After a converged solution is obtained, the usual normalization procedure can be followed. First, the real flux shape is normalized to yield the desired initial (transient state) reactor power. Then the adjoint flux is normalized to satisfy the quasi-static constraint condition,

$$\text{constant} = \langle \psi^*, \frac{1}{V} \psi \rangle . \quad (\text{H-17})$$

The normalized flux still satisfies the spatial equation because Eq. (H-16) is independent of the scalar magnitude of ψ .

APPENDIX I

DERIVATION AND CALCULATIVE APPROACH FOR THE MULTIGROUP NEUTRON DIFFUSION EQUATIONS

The multigroup diffusion equations solved by SIMMER-II are given in Chap. III. This appendix contains a derivation of the corresponding difference equations. These equations then are cast into matrix form, and the calculative expressions are defined. The matrix solution for the spatial flux is obtained by standard methods.

The geometry is assumed to be cylindrical with a nonuniform mesh. The radial mesh is numbered from 1 to IM, with mesh-cell centers indicated by subscript "i." Cell boundaries are indicated on the left by "i-1/2" and on the right by "i+1/2." The axial mesh is numbered from 1 to JM, with mesh-cell centers indicated by subscript "j." Axial cell boundaries are indicated on the bottom by "j-1/2" and on the top by "j+1/2."

I. DERIVATION OF THE DIFFUSION FINITE-DIFFERENCE EQUATIONS

The energy-dependent neutron diffusion equation [Eq. (III-259)] is repeated here for convenience.

$$\begin{aligned} \frac{1}{V(E)} \frac{\partial \Phi}{\partial t} - \frac{1}{r} \frac{\partial}{\partial r} \left(D r \frac{\partial \Phi}{\partial r} \right) - \frac{\partial}{\partial z} \left(D \frac{\partial \Phi}{\partial z} \right) + \Sigma_t \Phi = \int_E^\infty \Sigma_s(t, r, z, E' \rightarrow E) \Phi(t, r, z, E') dE' \\ + \frac{x_p(E)}{k_0} \int_0^\infty \nu_p \Sigma_f(t, r, z, E') \Phi(t, r, z, E') dE' + \sum_{d=1}^{\text{IGD}} x_d(E) \lambda_d C_d. \end{aligned} \quad (\text{I-1})$$

where

$\Phi = \Phi(t, r, z, E)$ is the neutron flux per unit volume and per unit energy at the spatial point (r,z) with energy E at time t,

$V(E)$ is the neutron velocity,

$\Sigma_t = \Sigma_t(t, r, z, E)$ is the macroscopic total cross section,

$\Sigma_s(t, r, z, E' \rightarrow E)$ is the macroscopic cross section for isotropic neutron scattering from neutron energy E' to neutron energy E,

$\Phi(t,r,z,E)$ is the neutron scalar flux,

k_0 is the initial stationary eigenvalue for the reactor system,

$\chi_p(E)$ is the prompt neutron fission emission spectrum,

ν_p is the prompt neutron fission yield,

$\Sigma_f(t,r,z,E)$ is the macroscopic fission cross section,

IGD is the number of delayed-neutron precursor groups,

$\chi_d(E)$ is the delayed-neutron emission spectrum from delayed-neutron precursor group d,

λ_d is the decay constant for delayed-neutron precursor group d, and

$C_d = C_d(t,r,z)$ is the delayed-neutron precursor concentration for precursor group d.

The multigroup neutron diffusion equation is formed by applying the operator $\int_{\Delta E_g} (\cdot) dE$ to Eq. (I-1). This yields

$$\begin{aligned} \frac{1}{V_g} \frac{\partial \Phi_g}{\partial t} - \frac{1}{r} \frac{\partial}{\partial r} \left(D_g r \frac{\partial \Phi_g}{\partial r} \right) - \frac{\partial}{\partial z} \left(D_g \frac{\partial \Phi_g}{\partial z} \right) + \Sigma_R \Phi_g \\ = \sum_{g'=1}^{g-1} \Sigma_{s,g' \rightarrow g} \Phi_{g'} + \frac{\chi_p}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{g'} \Phi_{g'} + \sum_{d=1}^{IGD} \chi_{dg} \lambda_d C_d \quad . \quad (I-2) \end{aligned}$$

In Eq. (I-2), $\Phi_g = \int_{\Delta E_g} \Phi(t,r,z,E) dE$, and Σ_s is averaged in a way that preserves reaction rates. The variables Σ_R are the removal cross sections (denoting the removal of neutrons from group g by absorption or downscatter). Note that

$$\Sigma_{t,g} = \Sigma_{R,g} + \Sigma_{s,g \rightarrow g} \quad .$$

The spatial variables are averaged by applying the operator

$$\bar{O}(\cdot) = \int_{z_{j-1/2}}^{z_{j+1/2}} \int_{r_{i-1/2}}^{r_{i+1/2}} 2\pi r dr dz$$

with the associated definitions.

1. Axial mesh size: $\Delta z_j = z_{j+1/2} - z_{j-1/2}$
2. Radial mesh size: $\Delta r_i = 2(r_{i+1/2} - r_i) = r_{i+1/2} - r_{i-1/2}$
3. Cell volume: $V_{ij} = \bar{O}(1) = \pi(r_{i+1/2}^2 - r_{i-1/2}^2) \Delta z_j$

$$= 2\pi r_i \Delta r_i \Delta z_j$$
4. Average flux: $\phi_{ijg} = \frac{\bar{O}(\phi_g)}{\bar{O}(1)} = \frac{\bar{O}(\phi_g)}{V_{ij}}$

Applying operator $\bar{O}$ to Eq. (I-2) yields

$$\begin{aligned}
 \frac{V_{ij}}{V_g} \frac{\partial \phi_{ijg}}{\partial t} &= 2\pi r_{i+1/2} \Delta z_j D_{ijg} \left(\frac{\partial \phi_g}{\partial r} \right) \Big|_{r_{i+1/2}} + 2\pi r_{i-1/2} \Delta z_j D_{ijg} \left(\frac{\partial \phi_g}{\partial r} \right) \Big|_{r_{i-1/2}} \\
 &\quad - 2\pi r_i \Delta r_i D_{ijg} \left(\frac{\partial \phi_g}{\partial z} \right) \Big|_{z_{j+1/2}} + 2\pi r_i \Delta r_i D_{ijg} \left(\frac{\partial \phi_g}{\partial z} \right) \Big|_{z_{j-1/2}} \\
 &\quad + V_{ij} \Sigma_{R,ijg} \phi_{ijg} = V_{ij} \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} \\
 &\quad + V_{ij} \frac{\chi_{pg}}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} + V_{ij} \sum_{d=1}^{IGD} \chi_{dg} \lambda_d C_{ijd} \quad . \quad (I-3)
 \end{aligned}$$

The radial flux derivatives (the radial leakage terms) are evaluated at the mesh-cell boundaries in terms of the radial neutron current ($D \partial \phi_g / \partial r$); that is,

$$\begin{aligned}
 \int \frac{1}{r} \frac{\partial}{\partial r} (D_g r \frac{\partial \phi_g}{\partial r}) r dr &= r_{i+1/2} D_{ijg} \frac{\partial \phi_g}{\partial r} \Big|_{r_{i+1/2}} \\
 &\quad - r_{i-1/2} D_{ijg} \frac{\partial \phi_g}{\partial r} \Big|_{r_{i-1/2}} \quad . \quad (I-4)
 \end{aligned}$$

Assuming continuity of flux and current at the radial mesh-cell boundaries yields an expression for the mesh-cell boundary flux.

For the right mesh-cell boundary, continuity of current is expressed as

$$D_{i+1jg} \frac{\Phi_{i+1jg} - \Phi_{i+1/2jg}}{r_{i+1} - r_{i+1/2}} = D_{ijg} \frac{\Phi_{i+1/2jg} - \Phi_{ijg}}{r_{i+1/2} - r_i} , \quad (I-5)$$

where continuity of flux is implicit because of the single boundary flux, $\Phi_{i+1/2jg}$. The left mesh-cell boundary yields a similar expression. Solving for the right mesh-cell boundary flux yields

$$\Phi_{i+1/2jg} = \frac{D_{i+1jg} \Delta r_i \Phi_{i+1} + D_{ijg} \Delta r_{i+1} \Phi_{ijg}}{D_{i+1jg} \Delta r_i + D_{ijg} \Delta r_{i+1}} .$$

Substituting the result into the leakage expression [Eq. (I-4)] yields

$$r_{i+1/2} D_{ijg} \frac{\partial \Phi_g}{\partial r} \Big|_{r_{i+1/2}} = r_{i+1/2} \frac{2D_{i+1jg} D_{ijg}}{\Delta r_i D_{i+1jg} + \Delta r_{i+1} D_{ijg}} (\Phi_{i+1jg} - \Phi_{ijg}) ,$$

and (I-6)

$$r_{i-1/2} D_{ijg} \frac{\partial \Phi_g}{\partial r} \Big|_{r_{i-1/2}} = r_{i-1/2} \frac{2D_{i-1jg} D_{ijg}}{\Delta r_i D_{i-1jg} + \Delta r_{i-1} D_{ijg}} (\Phi_{ijg} - \Phi_{i-1jg}) .$$

Similarly, the axial leakage terms are given by

$$\int \frac{\partial}{\partial z} (D_g \frac{\partial \Phi_g}{\partial z}) dz = D_{ijg} \frac{\partial \Phi_g}{\partial z} \Big|_{z_{j+1/2}} - D_{ijg} \frac{\partial \Phi_g}{\partial z} \Big|_{z_{j-1/2}} . \quad (I-7)$$

Again, continuity of flux and current at mesh-cell boundaries yields an expression for the mesh-cell boundary fluxes. The upper boundary current continuity equation is expressed in finite-difference form as

$$D_{ij+1g} \frac{\phi_{ij+1g} - \phi_{ij+1/2g}}{z_{j+1} - z_{j+1/2}} = D_{ijg} \frac{\phi_{ij+1/2g} - \phi_{ijg}}{z_{j+1/2} - z_j}, \quad (I-8)$$

where continuity of flux is implicit because of the single boundary flux, $\phi_{ij+1/2g}$. The lower-cell boundary yields a similar expression. Solving for the upper mesh-cell boundary flux yields

$$\phi_{ij+1/2g} = \frac{D_{ij+1g} \Delta z_j \phi_{ij+1g} + D_{ijg} \Delta z_{j+1} \phi_{ijg}}{D_{ij+1g} \Delta z_j + D_{ijg} \Delta z_{j+1}}.$$

Substitution of the result into the leakage expression [Eq. (I-7)] yields

$$D_{ijg} \frac{\partial \phi_g}{\partial z} \Big|_{z_{j+1/2}} = \frac{2D_{ijg} D_{ij+1g}}{\Delta z_j D_{ij+1g} + \Delta z_{j+1} D_{ijg}} (\phi_{ij+1g} - \phi_{ijg}) \quad (I-9)$$

and

$$D_{ijg} \frac{\partial \phi_g}{\partial z} \Big|_{z_{j-1/2}} = \frac{2D_{ij-1g} D_{ijg}}{\Delta z_{j-1} D_{ijg} + \Delta z_j D_{ij-1g}} (\phi_{ijg} - \phi_{ij-1g}).$$

Substituting Eqs. (I-6) and (I-9) into Eq. (I-3) yields

$$\begin{aligned} \frac{V_{ij}}{V_g} \frac{\partial \phi_{ijg}}{\partial t} - 2\pi \Delta z_j \left[r_{i+1/2} \frac{2D_{i+1jg} D_{ijg}}{\Delta r_i D_{i+1jg} + \Delta r_{i+1} D_{ijg}} (\phi_{i+1jg} - \phi_{ijg}) \right. \\ \left. - r_{i-1/2} \frac{2D_{i-1jg} D_{ijg}}{\Delta r_{i-1} D_{ijg} + \Delta r_i D_{i-1jg}} (\phi_{ijg} - \phi_{i-1jg}) \right] \end{aligned}$$

$$\begin{aligned}
& - \pi(r_{i+1/2}^2 - r_{i-1/2}^2) \left[\frac{2D_{ijg}D_{ij+1g}}{\Delta z_j D_{ij+1g} + \Delta z_{j+1} D_{ijg}} (\phi_{ij+1g} - \phi_{ijg}) \right. \\
& \left. - \frac{2D_{ij-1g}D_{ijg}}{\Delta z_{j-1} D_{ijg} + \Delta z_j D_{ij-1g}} (\phi_{ijg} - \phi_{ij-1g}) \right] \\
& + v_{ij} \Sigma_{R,ijg} \phi_{ijg} = v_{ij} \left(\frac{\chi_{pg}}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} \right. \\
& \left. + \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} + \sum_{d=1}^{IGD} \chi_{dg} \lambda_d C_{ijd} \right) . \tag{I-10}
\end{aligned}$$

Equation (I-10) is a finite-difference equation in space and energy for the time-dependent diffusion equation; however, the time derivative is not expressed in finite-difference form. In SIMMER-II, the quasi-static method is used to treat the time dependence; this method is explained in detail in Sec. V.C. However, the flux factorization (upon which the quasi-static method is based) is introduced to obtain the form for the time dependence that appears in the SIMMER-II flux-shape solution modules.

In the quasi-static method, the time-dependent flux is factored into an amplitude function, $N(t)$, and a flux shape, $\phi(t,r,z,E)$; that is,

$$\Phi(t,r,z,E) = N(t) \phi(t,r,z,E) , \tag{I-11}$$

where N contains most of the time dependence and the flux shape varies only slowly in comparison. The flux equation is split into two equations: an ordinary differential equation set for N , which is solved easily, and the flux-shape equation, which is solved only when significant changes in the flux shape have occurred.

Substituting Eq. (I-11) in the first term of Eq. (I-10) yields

$$\frac{v_{ij}}{v_g} \frac{\partial \phi_{ijg}}{\partial t} = N \frac{v_{ij}}{v_g} \frac{\partial \phi_{ijg}}{\partial t} + \phi_{ijg} \frac{v_{ij}}{v_g} \frac{dN}{dt} . \tag{I-12}$$

The partial derivative of the flux shape is replaced by a backward-difference relation; that is,

$$\frac{\partial \phi_{ijg}}{\partial t} = \frac{\phi_{ijg} - \phi_{ijg}^{k-1}}{\Delta t} \quad (I-13)$$

where the k-1 superscript indicates the flux shape at $t-\Delta t$, which is the previous time step. Because $\frac{dN}{dt}$ is replaced by a numerical value determined outside the flux-shape solution (in the amplitude equation solution), it is left in differential form. Substituting Eqs. (I-11) and (I-13) into Eq. (I-10) yields, after division by N,

$$\begin{aligned} & - 2\pi \Delta z_j \left[r_{i+1/2} \frac{2 D_{i+1jg} D_{ijg}}{\Delta r_i D_{i+1jg} + \Delta r_{i+1} D_{ijg}} (\phi_{i+1jg} - \phi_{ijg}) \right. \\ & \left. - r_{i-1/2} \frac{2 D_{i-1jg} D_{ijg}}{\Delta r_{i-1} D_{ijg} + \Delta r_i D_{i-1jg}} (\phi_{ijg} - \phi_{i-1jg}) \right] \\ & - \pi (r_{i+1/2}^2 - r_{i-1/2}^2) \left[\frac{2 D_{ijg} D_{ij+1g}}{\Delta z_j D_{ij+1g} + \Delta z_{j+1} D_{ijg}} (\phi_{ij+1g} - \phi_{ijg}) \right. \\ & \left. - \frac{2 D_{ij-1g} D_{ijg}}{\Delta z_{j-1} D_{ijg} + \Delta z_j D_{ij-1g}} (\phi_{ijg} - \phi_{ij-1g}) \right] \\ & + v_{ij} (\Sigma_{r,ijg} + \frac{1}{v_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{v_g} \frac{1}{\Delta t}) \phi_{ijg} \\ & = v_{ij} \left(\frac{x_{pg}}{k_o} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} + \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} \right. \\ & \left. + \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd} + \frac{1}{v_g} \frac{1}{\Delta t} \phi_{ijg}^{k-1} \right) \quad (I-14) \end{aligned}$$

II. BOUNDARY CONDITIONS

Equation (I-14) applies to the interior region of the spatial mesh. For the boundary mesh cells, the leakage terms must be modified to satisfy boundary conditions. In SIMMER-II, the boundary conditions are fixed; a zero current condition is assumed for the left boundary, and a zero flux condition is assumed for the other three boundaries.

For the left boundary, the second term on the right of Eq. (I-4) is zero; that is, there is no leakage to the left. Thus, using the flux shape instead of the flux,

$$\int \frac{1}{r} \frac{\partial}{\partial r} (D_g r \frac{\partial \phi_g}{\partial r}) r dr = r_{3/2} D_{2jg} \frac{\partial \phi_g}{\partial r} \Big|_{r_{3/2}} \quad (I-15)$$

The evaluation of the leakage then proceeds as in Eqs. (I-5) and (I-6), with the result that the second term in the first bracketed expression in Eq. (I-10) is eliminated.

For the right boundary, the leakage is given by

$$\begin{aligned} \int \frac{1}{r} \frac{\partial}{\partial r} (D_g r \frac{\partial \phi_g}{\partial r}) r dr &= r_{IM+1/2} D_{IMjg} \frac{\partial \phi_g}{\partial r} \Big|_{r_{IM+1/2}} \\ &- r_{IM-1/2} D_{IM-1jg} \frac{\partial \phi_g}{\partial r} \Big|_{r_{IM-1/2}} \quad , \end{aligned} \quad (I-16)$$

again using Eq. (I-4) with the flux shape substituted for the flux. In finite-difference form, the first term of Eq. (I-16) is

$$r_{IM+1/2} D_{IMjg} \frac{\partial \phi_g}{\partial r} \Big|_{r_{IM+1/2}} = r_{IM+1/2} D_{IMjg} \frac{-\phi_{IMjg}}{r_{IM+1/2} - r_{IM}} \quad , \quad (I-17)$$

using $\phi_{IM+1/2} = 0$ as required.

The leakage on the mesh-cell left boundary is evaluated as described previously.

As a result of Eq. (I-17), the first term in the first bracketed expression in Eq. (I-10) becomes

$$-r_{IM+1/2} \frac{2D_{IMjg}}{\Delta r_{IM}} \phi_{IMjg} \quad (I-18)$$

For the upper boundary, the leakage is given by Eq. (I-7) with the flux shape substituted for the flux:

$$\int \frac{\partial}{\partial z} (D_g \frac{\partial \phi_g}{\partial z}) dz = D_{iJMg} \frac{\partial \phi_g}{\partial z} \Big|_{z_{JM+1/2}} - D_{iJM-1g} \frac{\partial \phi_g}{\partial z} \Big|_{z_{JM-1/2}} \quad (I-19)$$

In finite-difference form, the first term is approximated as

$$D_{iJMg} \frac{\partial \phi_g}{\partial z} \Big|_{z_{JM+1/2}} = D_{iJMg} \frac{-\phi_{iJMg}}{z_{JM+1/2} - z_{JM}} \quad (I-20)$$

using $\phi_{iJM+1/2g} = 0$. The leakage on the mesh-cell upper boundary is evaluated as in Eq. (I-9). As a result of Eq. (I-20), the first term in the second bracketed expression in Eq. (I-10) becomes

$$- \frac{2D_{iJMg}}{\Delta z_{JM}} \phi_{iJMg} \quad (I-21)$$

For the lower boundary, the leakage also is given by Eq. (I-7) with the flux shape substituted for the flux:

$$\int \frac{\partial}{\partial z} (D_g \frac{\partial \phi_g}{\partial z}) dz = D_{i2g} \frac{\partial \phi_g}{\partial z} \Big|_{z_{3/2}} - D_{i1g} \frac{\partial \phi_g}{\partial z} \Big|_{z_{1/2}} \quad (I-22)$$

In finite-difference form, the second term is approximated by

$$D_{i1g} \frac{\partial \phi_g}{\partial z} \Big|_{z_{1/2}} = D_{i1g} \frac{-\phi_{i1g}}{z_1 - z_{1/2}} \quad , \quad (I-23)$$

using $\phi_{i1/2g} = 0$. The leakage on the mesh-cell upper boundary is evaluated as in Eq. (I-9). As a result of Eq. (I-23), the second term in the second bracketed expression in Eq. (I-10) becomes

$$- \frac{2D_{i1g}}{\Delta z_1} \phi_{i1g} \quad . \quad (I-24)$$

III. SIMMER-II CALCULATIVE EXPRESSIONS FOR THE DIFFUSION EQUATION

Equation (I-14) and the boundary terms [Eqs. (I-18), (I-21), and (I-24)] form a set of $IM * JM$ equations for each energy group. In matrix form, the familiar five-diagonal band structure is obtained by numbering the unknowns, ϕ_{ijg} , sequentially in order of increasing i , then in order of increasing j . The diagonal vectors consisting of the nonzero entries in the coefficient matrix have elements defined by Eq. (I-14) and the boundary conditions. By collecting terms in Eq. (I-14), the following matrix elements are identified.

- (1) The upper super-diagonal vector consists of coefficients used to obtain the leakage to the right of a given spatial cell. Those coefficients, R_K , multiply the fluxes ϕ_{i+1jg} (except for the rightmost column), and ϕ_{ijg} . The elements R_K , which multiply ϕ_{i+1jg} , are defined by

$$R_K = -2\pi\Delta z_j r_{i+1/2} \frac{2D_{i+1jg}D_{ijg}}{\Delta r_i D_{i+1jg} + \Delta r_{i+1} D_{ijg}} \quad ,$$

$$\text{for } K = (j-1)*IM + (i+1), \text{ for } i=1, \dots, (IM-1), \text{ and all } j. \quad (I-25)$$

The boundary values for the rightmost column are given by

$$R_K = 2\pi\Delta z_j r_{IM+1/2}^{IM} \frac{2D_{IMjg}}{\Delta r_{IM}} , \text{ for } K = (j-1)*IM, \text{ for all } j .$$

Thus, coefficients for the rightmost column, ϕ_{ijg} , are located at the beginning of each row. This vector has length $IM * JM$.

- (2) The lower super-diagonal vector consists of coefficients used to obtain the leakage to the left of a given spatial cell. These coefficients multiply the fluxes ϕ_{i-1jg} (except for the leftmost column) and ϕ_{ijg} , respectively. The elements L_K are defined by

$$L_K = 0 , \text{ for } K = (j-1)*IM + 1 , \quad (I-26)$$

and

$$L_K = -2\pi\Delta z_j r_{i-1/2}^{i-1} \frac{2D_{i-1jg}D_{ijg}}{\Delta r_{i-1}D_{ijg} + \Delta r_i D_{i-1jg}}$$

$$\text{for } K = (j-1)*IM + i+1 .$$

This vector has length $IM * JM$. Note that, except for the boundary terms, $L_{K+1} = R_K$; thus, they are neither calculated separately nor stored.

- (3) The upper off-diagonal vector consists of coefficients to obtain the leakage to the top of a given spatial cell. It multiplies the fluxes ϕ_{ij+1g} (except for the top row) and ϕ_{ijg} . The elements U_K that multiply ϕ_{ij+1g} are defined by

$$U_K = -\pi(r_{i+1/2}^2 - r_{i-1/2}^2) \frac{2D_{ijg}D_{ij+1g}}{\Delta z_j D_{ij+1g} + \Delta z_{j+1} D_{ijg}} ,$$

$$\text{for } K = (j-1)*IM + i, \text{ for all } i, \text{ and } j=1, \dots, (JM-1) .$$

The boundary values for the upper row are given by

$$U_K = -\pi(r_{i+1/2}^2 - r_{i-1/2}^2) \frac{2D_{iJM}g}{\Delta z_{JM}} ,$$

$$\text{for } K = (JM-1)*IM + i, \text{ for all } i . \quad (I-27)$$

This vector has length $IM * JM$.

- (4) The lower off-diagonal vector consists of coefficients to obtain the leakage to the bottom of a given spatial cell and thus multiplies fluxes ϕ_{ij-1g} (except for the bottom row) and ϕ_{ijg} . The elements B_K are defined by

$$B_K = -\pi(r_{i+1/2}^2 - r_{i-1/2}^2) \frac{2D_{i1g}}{\Delta z_1} , \quad \text{for } K = i , \quad (I-28)$$

and

$$B_K = -\pi(r_{i+1/2}^2 - r_{i-1/2}^2) \frac{2D_{ij-1g}D_{ijg}}{\Delta z_{j-1}D_{ijg} + \Delta z_j D_{ij-1g}} ,$$

$$\text{for } K = (j-1)*IM + i .$$

Note that, except for the boundary terms, $B_{K+IM} = U_K$; thus, they are neither calculated separately nor stored. This vector has length $IM * JM$. However, the boundary terms $B_K, K = 1, \dots, IM$ are calculated and stored by at the beginning of the U_K 's with the U_K 's transposed by one row.

- (5) The main diagonal vector consists of coefficients to obtain the leakage through the four cell boundaries and the in-cell change terms and thus multiplies fluxes ϕ_{ijg} . The elements M_K are defined by

$$M_K = V_{ij} \left(\Sigma_{R,ijg} + \frac{1}{V_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{V_g} \frac{1}{\Delta t} \right) + R_K + L_K + U_K + B_K , \quad (I-29)$$

where $K = (j-1)*IM+i$. This vector has length $IM * JM$. The source vector also is defined by Eq. (I-14):

$$S_K = v_{ij} \left(\frac{x_{pg}}{k_0} \sum_{g'=1}^{IGM} (v_p \Sigma_f)_{ijg'} \phi_{ijg'} + \sum_{g'=1}^{g-1} \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} \right. \\ \left. + \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd} + \frac{1}{v_g} \frac{1}{\Delta t} \phi_{ijg}^{k-1} \right) \quad (I-30)$$

This vector has length $IM * JM$.

Because the matrix coefficients for a given energy group do not change during the outer iteration of the flux solution algorithm, the matrix elements are calculated as the cross sections are determined. The coefficients then are stored and accessed as needed during the flux iteration. However, the source vector is recalculated after each outer iteration and therefore cannot be treated similarly. In the following, the negative signs in front of the coefficient terms have been moved to the matrix equation solution. They are written on the left side of the equation below to preserve the previous definitions. Using the definition $D = \frac{1}{3\Sigma_{tr}}$ and rearranging Eqs. (I-25) yields

$$-R_K = \frac{(2/3)(\Delta z_j)(2\pi r_{i+1/2})}{\Delta r_i \Sigma_{tr,ijg} + \Delta r_{i+1} \Sigma_{tr,i+1jg}} \quad , \text{ for } K = (j-1)*IM(i+1) \quad ,$$

for $i = 1, \dots, (IM-1)$, and all j and

$$-R_K = \frac{(2/3)(\Delta z_j)(2\pi r_{IM+1/2})}{\Delta r_{IM} \Sigma_{tr,IMjg}} \quad , \text{ for } K = (j-1)*IM+1 \text{ for all } j.$$

Substituting the terms defined in Table I-I, leads to

$$-R_K = \frac{(2/3)A(LYH+j-1)*A(LA4+i)}{A(LA6+i-1)*\Sigma_{tr,ijg} + A(LA6+i)*\Sigma_{tr,i+1jg}} \quad ,$$

for $K = (j-1)*IM + (i+1)$, for $i = 1, \dots, (IM-1)$, and all j ,

and

$$-R_K = \frac{(2/3)A(LYH+j-1)*A(LA4+IM)}{A(LA6+IM-1)*\sum_{tr, IMjg}}, \text{ for } K = (j-1)*IM+1 \quad (I-31)$$

Using the definition $D = \frac{1}{\sum_{tr}}$ and rearranging Eq. (I-27) yields

$$-U_{K+IM} = \frac{(2/3) [\pi(r_{i+1/2}^2 - r_{i-1/2}^2)]}{\Delta z_j \sum_{tr, ijg} + \Delta z_{j+1} \sum_{tr, ij+1g}} \text{ for } K = (j-1)*IM + i, \text{ for all } i,$$

TABLE I-I
DEFINITION AND LOCATION OF CALCULATIVE
VARIABLES USED FOR DIFFUSION THEORY

<u>Storage Location Pointer Name</u>	<u>Length of Array</u>	<u>Mathematical Terms</u>
LA1	IM	$2\pi(r_{i+1/2} + r_{i-1/2})$
LA2	IM	$2\pi(r_{i+1/2} - r_{i-1/2})$
LA3	IM	$\pi(r_{i+1/2}^2 - r_{i-1/2}^2)$
LA4	IM + 1	$2\pi r_{i-1/2}$
LA5	IM	$\pi(r_{i+1/2}^2 - r_{i-1/2}^2)$
LA6	IM	$r_{i+1/2} - r_{i-1/2}$
LB1	JM	$\frac{1}{z_{j+1/2} - z_{j-1/2}}$
LYH	JM	$z_{j+1/2} - z_{j-1/2}$

and

$j = 1, \dots, (JM - 1)$, and

$$U_{K+IM} = \frac{(2/3) [\pi(r_{i+1/2}^2 - r_{i-1/2}^2)]}{\Delta z_{JM} \Sigma_{tr, iJMg}} \quad \text{for } K = (JM-1)*IM + i, \text{ for all } i,$$

where the U_K 's are transposed by one row (IM values) to permit the U_K 's to substitute for the B_K 's (using $U_K = B_{K+IM}$). An extra row is added at the beginning to provide for the needed B_1 through B_{IM} ; thus,

$$-U_K = \frac{(2/3) [\pi(r_{i+1/2}^2 - r_{i-1/2}^2)]}{\Delta z_1 \Sigma_{tr, i1g}}, \quad \text{for } K = i, i = 1, \dots, IM.$$

Substituting the terms defined in Table I-I leads to

$$-U_K = \frac{(2/3) A(LA3 + I - 1)}{A(LYH) \Sigma_{tr, i1g}}, \quad \text{for } K = i, i = 1, \dots, IM \quad \text{and}$$

$$-U_{K+IM} = \frac{(2/3) A(LA3+I-1)}{A(LYH+J-1) \Sigma_{tr, iJg} + A(LYH+J) \Sigma_{tr, iJ+1g}}$$

for $K = (j-1)*IM+i$, for all i , and $j = 1, \dots, (JM-1)$, and

$$-U_{K+IM} = \frac{(2/3) A(LA3+I-1)}{A(LYH+JM-1) \Sigma_{tr, iJg}}, \quad \text{for } K = (JM-1)*IM + i, \text{ for all } i. \quad (I-32)$$

The diagonal terms are generated from Eq. (I-29) using the U_K 's and R_K 's and the relations $R_K = L_{K+1}$ and $U_K = B_{K+IM}$.

APPENDIX J

SIMMER-II COMMON VARIABLE DICTIONARY

All FORTRAN variables except constants contained in the SIMMER-II common blocks are listed and defined in Table J-I. They are listed in alphabetical order for easy reference. Where appropriate, the mathematical symbol used in this report is listed along with the FORTRAN variable.

TABLE J-I

SIMMER-II COMMON BLOCK FORTRAN VARIABLES

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
A	SIMCOM (IA)		Large array used to contain neutronic array data. Data located by pointer system.
A0	COMINP	a_0	Donor-cell constant.
A1	KINET		$A1(I) = 1./I$, where $I = 1, \dots, 5$.
A12	RXCHA		Variable used as temporary storage in computing liquid-liquid heat transfer.
A1RHO	RXCHA	$a_{1,m}^K$	Defined by Eq. (IV-98).
A2RHO	RXCHA	$a_{2,m}^K$	Defined by Eq. (IV-99).
AA	SIMCOM (KOM)		Dummy word used as first word in common block SIMCOM(KOM).
AA	SIMCOM (IA)		$\mu_m(\frac{A_{i+1/2,j} + A_{i-1/2,j}}{\Delta z_j})$. Used only for the for the transport theory option (DIFF not defined).
AA1	KLC1, KLC2, KLC3		First word of the large-memory array, LCM, on the CDC-7600.
AA2	KLC2		The array in the KLC2 common block.
AA3	KLC3		The array in the KLC3 common block.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
AAROW	SIMCOM (KOM)		AAROW = IAAROW, an erasable array sufficiently large to contain all the region-wise data, the point kinetics data, the data being transferred from neutronics/fluid mechanics to fluid mechanics/neutronics, or the data being transferred to TAPE6 or TAPE36, or one row of the fluid-dynamics variables.
AASC	SIMCOM (KOM)		Small-core array that can contain three rows of fluid-dynamics variables.
AB	SOLVE		Array containing the bottom matrix elements used in the call to subroutine MATSOL.
ACAN	SIMCOM (KOM)	A_{can}^i	Input value of the can-wall surface area per unit volume for a mesh cell.
ACANW	RXCHA	A_{can}	Final value of the can-wall surface area per unit volume for a mesh cell.
ACLAD	SIMLOM (KOM)	A_{clad}^i	The input cladding surface area per unit volume in a mesh cell.
ACT	RXCHA		Temporary storage for the symmetric matrix described by Eq. (IV-70) to be used if a recalculation is required.
AD	SOLVE		Array containing the diagonal matrix elements used in the call to subroutine MATSOL.
ADT	SIMCOM (KOM)		Temporary storage used in the AIM algorithm (Appendix G). Initially contains the AD array.
ADHERR	SIMCOM (KOM)	$2\Gamma_{LS,i+1/2}$ $\cdot \Delta r_{i+1/2}$	Loss flux of mass from the liquid field to the structure field, summed over two radial cells.
ADHERZ	SIMCOM (KOM)	$2\Gamma_{LS,j+1/2}$ $\cdot \Delta z_{j+1/2}$	Loss flux of mass from the liquid field to the structure field, summed over two axial cells.
AEOS10	SIMCOM (KOM)	a_{M10}	Fitting parameter for the gas constant, R_M .

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
AEOS11	SIMCOM (KOM)	a_{M11}	Fitting parameter for the gas constant, R_M .
AEOS20	SIMCOM (KOM)	a_{M20}	Fitting parameter for the gas constant, R_M .
AEOS21	SIMCOM (KOM)	a_{M21}	Fitting parameter for the gas constant, R_M .
AEOS30	SIMCOM (KOM)	a_{M30}	Fitting parameter for the gas constant, R_M .
AEOS31	SIMCOM (KOM)	a_{M31}	Fitting parameter for the gas constant, R_M .
AEOSLM	SIMCOM (KOM)	$a_{L,M}$	AEOS fitting parameter defined to obtain continuity in the liquid condensate energy at $T_{Sat,M} = 2/3 T_{Crt,M}$.
AFLOW	RXCHA	A_{flow}	Fluid flow area per unit volume used in the structure configuration model. Initially defined by Eq. (III-98) and then reset as the model integrates the presence of the various structural components.
AFLOWP	RXCHA		Temporary variable defined by AFLOW-AFROZ.
AFPOAF	RXCHA	$(A_{flow}/A_{flow}^i)^{1/2}$	Temporary variable in the structure configuration model.
AFROZ	RXCHA	$\alpha_{froz}/\Delta z_j$	Flow area per unit volume occupied by excess frozen material.
AKINFI	SIMCOM (KOM)		Large number (10^{10}) used for coupling fields together when large drag or friction is desired.
AL	SOLVE		Array containing the left-side matrix elements used in the call to subroutine MATSOL.
ALAM	KINET	λ_d	Delayed neutron decay constant for delayed family "d".
ALCM	KLC1		The array in the KLC1 common block.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ALCSCO	SIMCOM (KOM)	$\alpha_{G\Gamma}$	The vapor fraction of a cell's fluid volume, which must be exceeded to allow vapor condensation on the structure.
ALCVGM	SIMCOM (KOM)	$\ln R_{Mi}$	Logarithm of the infinitely dilute gas constant. See discussion following Eq. (III-55).
ALDRG	SIMCOM (KOM)		The structure-field volume fraction above which structure drag is set to AKINFI.
ALNFCW	SIMCOM (KOM)	$\alpha_{nf,cw}$	The nonflow volume fraction associated with the can wall. Included until radial motion is allowed between SIMMER rings.
ALNOFL	SIMCOM (KOM)	α_{nf}	Volume fraction not available for fluid flow until the cladding fails.
ALOR	RXCHA	$\sum_{m=1}^{NCLE} \alpha_{Lm}/r_{pm}$	Temporary variable to transfer information between subroutine TSHTR and subroutine FAM.
ALOR2	RXCHA	$\sum_{m=1}^{NCLE} \alpha_{Lm}/r_{pm}^2$	Temporary variable to transfer information between subroutine TSHTR and subroutine FAM.
ALPHO	SIMCOM (KOM)	α_o	The value of $\alpha_o(1 - \alpha_S)$ is the vapor-field volume fraction at the two-phase/single-phase transition.
ALPHG	SIMCOM (KOM)	α_G	Vapor-field volume fraction for a mesh cell.
ALPHGI	NITIAL	α_G^i	The initial vapor-field volume fraction used in the problem setup.
ALPHGP	RXCHA	$\alpha_G/(1 - \alpha_S)$	Temporary variable used in evaluation of Eq. (III-135).
ALPHS	SIMCOM (KOM)	α_S	Structure-field volume fraction for a mesh cell.
ALPHSQ	RXCHA		The divisor in Eq. (III-168).

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ALPHSI	NITIAL	α_S^i	The initial structure-field volume fraction used in the problem setup.
ALPMIN	SIMCOM (KOM)		The input minimum liquid-field volume fraction.
ALRCRT	SIMCOM (KOM)	$\ln R_{Crt,M}$	The logarithm of the critical point gas constant. See discussion following Eq. (III-55).
ALSRAT	RXCHA	Λ_m	Contact coefficient defined by Eq. (III-129) used to obtain the liquid-structure surface area.
ALT	SIMCOM (KOM)		Temporary storage used in the AIM algorithm (Appendix G). Initially contains the AL array.
ALSTAB	PASS		Storage array for the previous row's advanced-time structure-field volume fraction.
ALTAB	PASS		Storage array for the previous row's advanced-time vapor-field volume fraction.
AMP	TIME	N	Most recent power amplitude, normalized to unity at time = 0.
AMPCO	TIME		Coefficients for quadratic fit to the natural logarithm of the power amplitude.
ANOFI	SIMCOM (KOM)		No-flow filler volume fraction used in the plug calculative algorithm.
AORFP	RXCHA	$\alpha_{L5/rp5}$	Fuel solid-particle variable used in the Weber number droplet radius calculation.
AORSP	RXCHA	$\alpha_{L6/rp6}$	Steel solid-particle variable used in the Weber number droplet radius calculation.
APARA	SIMCOM (KOM)	$a_{G,M}$	Input parameter used to increase the gas constant above the critical temperature.
APELL	SIMCOM (KOM)	A_{pell}^i	Fuel pellet surface area per unit volume as given in the region parameter input.
APIN	RXCHA	A_{pin}	Fuel pin surface area per unit volume.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
APLUG	SIMCOM (KOM)		Acceleration of the movable plug.
AR	SOLVE		Array containing the right-side matrix elements used in the call to subroutine MATSOL.
ARG	RXCHA		Exponent used in Eq. (IV-103) to correct the vapor-side heat-transfer coefficient for mass transfer.
ARGG	RXCHA	$-\bar{c}_{pG} \Gamma_{IGmLm}$	Iterative variable in the Newton-Raphson iteration for correcting the vapor-side heat-transfer coefficient for mass transfer.
ARGGP	RXCHA		Updated temporary value for ARGG.
AS1	RXCHA		Density rate of change per unit volume for a fuel crust.
AS2	RXCHA		Density rate of change per unit volume for fuel particles.
AS3	RXCHA		Density rate of change per unit volume for can wall.
ART	SIMCOM (KOM)		Temporary storage used in the AIM algorithm (Appendix G). Initially contains the AR array.
ASOUND	SIMCOM (KOM)	A	Input coefficient for the liquid fuel velocity of sound expression, Eqs. (III-64).
AST	RXCHA		Variable used for equilibrium mass freezing rates.
AT	SOLVE		Array containing the top matrix elements used in the call to subroutine MATSOL.
ATL	SIMCOM (KOM)		Temporary working array for the AIM algorithm.
ATOM	SIMCOM (KOM)	σ_m	The molecular diameter for vapor material M.
ATOT	RXCHA	A_{Tot}	The total structure surface area per unit volume in a mesh cell.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ATOTO	RXCHA	A'_{Tot}	The total structure surface area per unit volume in a mesh cell before refrozen material is added.
ATR	SIMCOM (KOM)		Temporary working array for the AIM algorithm.
AXMG	SIMCOM (KOM)	$-\Gamma_{g6}$	Total mass transfer rate to the vapor field component of fission gas.
AXMLI	SIMCOM (KOM)	$-\Gamma_{l1}$	Total mass transfer rate to the liquid fertile fuel.
AXMLR	SIMCOM (KOM)	$-\Gamma_{l2}$	Total mass transfer rate to the liquid fissile fuel.
B	SIMCOM (IA)		Radial geometry constant. B is equivalenced to BB and redefined for use in FIXUP.
BO	COMINP	b_0	Donor-cell constant.
BB	SIMCOM (IA)		Radial geometry constant. $B = \pi(r_{i+1/2}^2 - r_{i-1/2}^2)$. See Eq. (L-11d).
BCT	RXCHA		Temporary storage for the right-side vector described by Eq. (IV-70) to be used if a recalculation is required.
BD	SIMCOM (KOM)		Temporary variable used in the rebalance calculation of the AIM method.
BETAH	KINET	β_{hi}	Decay heat fraction used for URANUS option point kinetics.
BETAS	KINET	β	Summation of effective delayed neutron fractions over delayed families for URANUS option.
BF	SIMCOM (KOM)		Temporary variable used in the rebalance calculation of the AIM method.
BK	SIMCOM (KOM)		Temporary variable used in the rebalance calculation of the AIM method.
BL	SIMCOM (KOM)		Temporary variable used in the rebalance calculation of the AIM method.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
BODT	SIMCOM (KOM)	$b_o \Delta t$	The product of BO and DT.
BOUT	SIMCOM (KOM)		Set to 1 for a nonrigid lower boundary condition (IBOT $\neq$ 0).
BQ	SIMCOM (KOM)		Temporary variable used in the rebalance calculation of the AIM method.
BR	SIMCOM (KOM)		Temporary variable used in the rebalance calculation of the AIM method.
BS	SIMCOM (KOM)		Temporary variable used in the rebalance calculation of the AIM method.
BSOUND	SIMCOM (KOM)	B	Input exponent for the liquid fuel velocity of sound expression, Eq. (III-64).
C	KINET	c_d	End-of-time-step value for the concentration of delayed neutron precursor family "d" used in the point kinetics algorithm for the URANUS option.
C	SIMCOM (IA)		Angular geometry constant. C is equivalenced to CC and redefined for use in FIXUP.
CO	KINET	CO_d	Beginning-of-time-step value for the neutron precursor concentration for delayed family "d".
CAEOS	SIMCOM (KOM)		Convergence precision for the analytic EOS.
CC	SIMCOM (IA)		Angular redistribution constant, defined Eq. (L-38).
CDR	SIMCOM (KOM)	C_D	The form drag coefficient used in the SIMMER-II KACHINA liquid-vapor drag correlation.
CFLR	SIMCOM (KOM)		The maximum mass-transfer rate of failed structure to the liquid or vapor field before cladding failure.
CINLAS	COMINP		Last word of COMINP common block.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
CMU	SIMCOM (KOM)	$\mu_{Crt,m}$	The critical viscosity for each component computed from the law of corresponding states.
COAL	SIMCOM (KOM)	$\frac{w\bar{V}_m}{8}$	Coalescence coefficient for determining droplet radii.
COLOUR	PSEDIT		Array of colors for plotting.
COMPL	SIMCOM (KOM)	$c_{\ell m}^2$	Square of the sound speed for liquid-field density component m.
CON3	RXCHA		Collection of parameters used in the fission gas release formulation.
CONST	RXCHA		Temporary variable used in the determination of the DTL(8) time step control.
CONTWT	SIMCOM (KOM)	w_{cont}	Control material reactivity worth distribution.
COOLWT	SIMCOM (KOM)	w_{NA}	Coolant (sodium) reactivity worth distribution.
COREL(4,6)	SIMCOM (KOM)	$C_{LSkm}, C_{Gsk},$ $C_{GLk}, k=1,2,3,$ $4, m=1,2,3,4$	Heat-transfer correlation parameters for liquid-structure, vapor-structure, and liquid-heat transfer. Variable storage in this array is in the sequence indicated at the left for liquid-structure heat transfer. Locations 5 and 6 are for vapor-structure and liquid-vapor transfers.
COURNT	SIMCOM (KOM)	C_t	Courant stability condition multiplier.
CPDHFG	RXCHA	$\bar{c}_{pG}/h_{\ell g,M}$	Mass-averaged heat capacity at constant pressure divided by the heat of vaporization.
CPGAS	RXCHA	c_{pGM}	Heat capacity at constant pressure for vapor-field material component M.
CPGBAR	EOS	$\bar{c}_p$	Mass-averaged heat capacity at constant pressure.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
CQ	SIMCOM (KOM)		Dummy fluid-dynamics cell variable used to temporarily store a cell-dependent variable prior to plotting.
CSOUND	SIMCOM (KOM)	C	Input minimum for the liquid fuel velocity of sound.
CT	SIMCOM (IA)		Loss terms in transport difference equation. Defined by Eq. (L-38).
CURINS	SIMCOM (IA)		Quasistatic leakage integral evaluated at the beginning of a shape time step (for the transport option).
CURINT	SIMCOM (IA)	I^S	Current quasistatic leakage integral (for the transport option).
CVG	SIMCOM (KOM)	c_{vGM}	Constant-volume specific heat for vapor material M.
CVGBAR	EOS	$\bar{c}_{vG}$	Averaged constant-volume specific heat for the vapor field.
CVGGM1	SIMCOM (KOM)	$c_{vGM}(\gamma_M - 1)$	Gas constant for vapor material M.
CVL	SIMCOM (KOM)	c_{vLM}	Constant-volume specific heat for liquid material M.
CVLP	SIMCOM (KOM)	c'_{vLM}	Defined for material M by Eq. (III-40).
CVS	SIMCOM (KOM)	c_{vSM}	Constant-volume specific heat for structure material M.
CVSURF	RXCHA		Heat capacity for a solid or liquid component in the multicomponent phase-transition model.
D	SIMCOM (IA)		Angular redistribution constant used in FIXUP. D is equivalenced to DD.
DD	SIMCOM (IA)		Angular redistribution constant used in FIXUP. D is equivalenced to DD.
DELRHO	RXCHA		The fuel density change used in the Newton-Raphson iteration coupling the mass-continuity equation and the EOS.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
DELTAT	TIME	Δt^S	Current flux-shape time step.
DELTIN	TIME		Initial flux-shape time step.
DELTR	TIME		Initial reactivity time step.
DEN	RXCHA		Contains the denominator of Eq. (III-87).
DF	SIMCOM (KOM)		Temporary variable used in subroutine MATSOL.
DFDRG	RXCHA	$dF^k/d\rho_{Gm}^k$	Defined by Eq. (IV-102).
DFDX	RXCHA		The functional derivative used in the Newton-Raphson iteration defined by Eq. (E-12).
DFV	SIMCOM (KOM)		Variable for passing the time and cycle number to the fluid dynamics postprocessor file.
DGKGGR	SIMCOM (KOM)		A momentum coupling term defined by the AASC array (Appendix O) and used in the pressure iteration.
DGKGGT	SIMCOM (KOM)		A momentum coupling term defined by the AASC array (Appendix O) and used in the pressure iteration.
DGKGLR	SIMCOM (KOM)		A momentum coupling term defined by the AASC array (Appendix O) and used in the pressure iteration.
DGKGLT	SIMCOM (KOM)		A momentum coupling term defined by the AASC array (Appendix O) and used in the pressure iteration.
DGKLGR	SIMCOM (KOM)		A momentum coupling term defined by the AASC array (Appendix O) and used in the pressure iteration.
DGKLGT	SIMCOM (KOM)		A momentum coupling term defined by the AASC array (Appendix O) and used in the pressure iteration.
DGKLLR	SIMCOM (KOM)		A momentum coupling term defined by the AASC array (Appendix O) and used in the pressure iteration.

TABLE J-I (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
DGKLLT	SIMCOM (KOM)		A momentum coupling term defined by the AASC array (Appendix O) and used in the pressure iteration.
DH	RXCHA	D_h	Hydraulic diameter for a mesh cell.
DHBELL	SIMCOM (KOM)	$1/D_h$	Inverse hydraulic diameter.
DHINGE	SIMCOM (KOM)	D_{hc}^i	Hydraulic diameter for a region with intact geometry.
DHNCLD	SIMCOM (KOM)	D_{hf}^i	Hydraulic diameter for a region with fuel pellets having no cladding.
DHNPEL	SIMCOM (KOM)	D_{can}^i	Hydraulic diameter for a region with no fuel pins in the subassembly.
DHSDTS	EOS	$\frac{\partial h_{lg,M}}{\partial T_{Sat,M}}$	Derivative of the heat of vaporization with respect to the saturation temperature.
DHSTOR	SIMCOM (KOM)	D_h	Hydraulic diameter stored for a long print output cycle.
DHVP	RXCHA		The difference between the current and previous iterative droplet vapor-side heat transfer coefficients used in Eq. (IV-116).
DHVP2	RXCHA		The difference between the $\kappa-1$ and $\kappa-2$ iterative droplet vapor-side heat transfer coefficients used in Eq. (IV-116).
DHVS	RXCHA		The difference between the current and previous iterative structure vapor-side heat-transfer coefficients used in Eq. (IV-116).
DHVSP	RXCHA		The difference between the $\kappa-1$ and $\kappa-2$ iterative structure vapor-side heat-transfer coefficients used in Eq. (IV-116).
DJMAX	TIME		Fluid dynamics time step control based on maximum phase-transition rate.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
DK1	RXCHA		Temporary variable used in calculation of the droplet radius based on the Weber number.
DKHET	KINET	H_i	Decay heat power by decay heat group.
DKHETP	KINET		Previous decay heat power by decay heat group.
DKLAM	KINET	λ_{hi}	Decay constant for each decay heat group. Used in the URANUS option.
DKYLD	KINET	β_{hi}	Decay heat fraction for each decay group. Used in the URANUS option.
DKYLDs	KINET	β_h	Total fraction of power because of decay heating. Used in the URANUS option.
DLT	RXCHA	$e_{Li,q,M} - \tilde{e}_{Lm}^{n+1}$	Energy difference used in the calculation of material freezing.
DMASS	RXCHA	$\bar{\rho}_G^{k+1} - \bar{\rho}_G^n$	Vapor-field density difference resulting from vaporization/condensation.
DMASSC	RXCHA	$\bar{\rho}_{Gm}^{k+1} - \bar{\rho}_{Gm}^n$	Vapor-field component density difference resulting from vaporization/condensation.
DMPIN	UNITS		Name of file (TAPE8) for reading restart data.
DMPOUT	UNITS		Name of a family of files (TAPEB, TAPEC, TAPED, and so forth) for writing restart data.
DOPCO	KINET	D_o	Doppler coefficient. Used in the URANUS option.
DOPPWT	SIMCOM (KOM)	$(DW)_{i,j}$	Doppler weight distribution. Used in the URANUS option.
DP	SIMCOM (KOM)	$\Delta p_{i,j}$	The change in pressure during a pressure iteration.
DPDE	EOS		The change in pressure with respect to energy for the tabular EOS.
DPMAX	RXCHA		The maximum pressure difference between a cell and its neighbors for use in the DTL(11) time step control.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
DPOWMX	KINET		Maximum fractional power change in a fluid dynamics time step.
DPURSF	SIMCOM (KOM)		Pressure difference across the unrestructured fuel annulus.
DQ	SIMCOM (KOM)		Temporary variable used in the AIM algorithm (Appendix G).
DR	SIMCOM (KOM)	Δr_i	The radial mesh spacing.
DREAMX	KINET		Maximum reactivity change in a fluid dynamics time step.
DRDRG	SIMCOM (KOM)	D_M	Defined by Eq. (IV-8).
DRIP	SIMCOM (KOM)	$1/2 (\Delta r_i + \Delta r_{i+1})$	Width of the (i+1/2) momentum cell.
DROJAY	SIMCOM (KOM)		The maximum fractional change in the smeared vapor density because of vaporization during a time step.
DSDM	SIMCOM (KOM)		Droplet size distribution multiplier.
DT	TIME	Δt	$DT = DTKACH =$ most recent fluid dynamics time step.
DTALPH	TIME		Parameter used to transmit the DTL(4) time step control to subroutine HYDRO.
DTAMP	RXCHA	$\Delta \tilde{t} N$	The product of the estimated time step used for the exchange function calculation and the power amplitude.
DTDMP	TIME		The time intervals between restart dumps before time TDMP.
DTF	TIME	$\Delta t^r - t^S$	Time interval between the most recent shape calculation and the most recent reactivity calculation.
DTGAM	TIME		The transfer variable from XCHANJ to HYDRO to control the time step for liquid-vapor phase-transition phenomena.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
DTGAMA	RXCHA		The variable containing the DTGAM calculation for each cell. It is compared with DTGAM to obtain a maximum value for a computational mesh.
DTGNP	RXCHA		The difference between the current and previous iterative vapor temperatures used in Eq. (IV-116).
DTGNP2	RXCHA		The difference between the $\kappa-1$ and $\kappa-2$ iterative vapor temperatures used in Eq. (IV-116).
DTKACH	TIME	Δt	Most recent fluid-dynamics time step.
DTL	TIME		Estimated fluid-dynamics time step for the various time step controls.
DTMAX	TIME		Maximum fluid dynamics time step.
DTMIN	SIMCOM (KOM)		The minimum fluid dynamics time step allowed; an attempt by the problem to define a smaller time step will result in a program abort.
DTMPG	TIME		The transfer variable from XCHANJ to HYDRO to control the time step for vapor-field energy changes.
DTMPGU	RXCHA		The variable containing the DTMPG calculation for each cell. It is compared with DTMPG to obtain a maximum value for the computational mesh.
DTMPL	TIME		The transfer variable from EXFLUD to HYDRO to control the time step for liquid-field component energy changes.
DTMPM	TIME		Fluid dynamics time-step control based on maximum vapor-field energy change.
DTN	TIME	$\Delta \tilde{t}$	Estimated fluid dynamics time step.
DTNMAX	TIME		Maximum allowed flux-shape time step.
DTNSHP	SIMCOM (IA)	Δt_{Sg}	Estimated flux-shape time step based on flux tilting.

TABLE J-I (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
DTO	PSEDIT		Time at which long-print intervals are changed.
DT02	SIMCOM (KOM)	$\Delta t/2$	One-half the current fluid dynamics time step.
DTOC	PSEDIT		Time intervals for long prints of fluid dynamics data.
DTOCP	PSEDIT		Time interval for writing mesh cell data for postprocessing.
DTOP	PSEDIT		Times at which time intervals for writing postprocessing files are changed.
DTP	TIME	Δt^{n-1}	Previous fluid dynamics time step.
DTPN	TIME	Δt^{S-1}	Previous flux-shape time step.
DTPOS	TIME		Last value of fluid dynamics time step prior to the flux shape calculation.
DTPRDT	SIMCOM (KOM)	$\Delta t^{n-1}/\Delta t$	Equal to DTP/DT.
DTR	TIME	Δt^r	Current reactivity time step.
DTRMAX	TIME		Maximum allowed reactivity time step.
DTRN	TIME		Estimated next reactivity time step.
DTRP	TIME	Δt^{r-1}	Previous reactivity time step.
DTSDRG	EOS	$(\frac{\partial T_{Sat,M}}{\partial \rho_{Gm}})$	The derivative of the saturation temperature for the density of vapor-field component m.
DTSDTG	EOS	$(\frac{\partial T_{Sat,M}}{\partial T_G})$	The derivative of the saturation temperature for vapor temperature.
DTSTAR	TIME		Initial fluid dynamics time step.
DTSN	RXCHA		The difference between the current and previous iterate for the saturation temperature used in Eq. (IV-116).
DTSNP	RXCHA		The difference between the $\kappa-1$ and $\kappa-2$ iterate for the saturation temperature used in Eq. (IV-116).

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
DTT	RXCHA		Temporary variable which equals DTN in calculation of the exchange functions.
DVLMGN	RXCHA		Velocity increment in the Newton-Raphson iteration for computing a droplet radius using the Weber number criterion.
DTXX	DUM		Fluid dynamics time step passed to the point kinetics routines.
DZ	SIMCOM (KOM)	Δz_j	The mesh spacing for axial row "j".
DZIP	SIMCOM (KOM)	$1/2 (\Delta z_j + \Delta z_{j+1})$	The axial mesh spacing for momentum cell $\Delta z_{j+1/2}$.
E1	SIMCOM (KOM)	Q_{N1}	Specific power shape (W/kg) for fertile fuel. Defined on the neutronics mesh in a space-dependent kinetics problem.
E2	SIMCOM (KOM)	Q_{N2}	Specific power shape (W/kg) for fissile fuel. Defined on the neutronics mesh in a space-dependent kinetics problem.
E3	SIMCOM (KOM)	Q_{N3}	Specific power shape (W/kg) for stainless steel. Defined on the neutronics mesh in a space-dependent kinetics problem.
E4	SIMCOM (KOM)	Q_{N4}	Specific power shape (W/kg) for sodium. Defined on the neutronics mesh in a space-dependent kinetics problem.
E5	SIMCOM (KOM)	Q_{N5}	Specific power shape (W/kg) for control material. Defined on the neutronics mesh in a space-dependent kinetics problem.
EBETA	KINET	β_d	Delayed neutron fraction for delayed neutron precursor family "d".
EFAC	TIME	$F_1(\rho)$	Quasi-static method time-step reduction factor.
EFAC2	TIME	$f_2(\rho)$	Quasi-static method time-step reduction factor.
EIGEN	SIMCOM (IA)	$1/k_0$	Stationary-state eigenvalue.

TABLE J-I (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
EL	KINET	A	Prompt neutron generation time. Used in the URANUS option.
ELFILM	TIME		The cumulative liquid-field component specific energy change since the previous postprocessor dump.
ELMOV	TIME		The maximum liquid-field component N specific energy change between post-processor dumps.
EMFL	SIMCOM (KOM)		Total liquid fuel ejected into the RCB.
EMFP	SIMCOM (KOM)		Total particulate fuel ejected into the RCB.
EMFV	SIMCOM (KOM)		Total fuel vapor ejected into the RCB.
EMNA	SIMCOM (KOM)		Total sodium ejected into the RCB.
EMSS	SIMCOM (KOM)		Total steel ejected into the RCB.
EN1	SIMCOM (KOM)	Q_{N1}	Specific power shape (W/kg) for fertile fuel. Multiplication by AMP gives the total nuclear energy deposition rate. For the point kinetics or power-vs-time options, this is the initial rate. For the quasi-static method, this is updated at reactivity and shape steps.
EN2	SIMCOM (KOM)	Q_{N2}	Specific power shape (W/kg) for fissile fuel. Multiplication by AMP gives the total nuclear energy deposition rate. For the point kinetics or power-vs-time options, this is the initial rate. For the quasi-static method, this is updated at reactivity and shape steps.
EN3	SIMCOM (KOM)	Q_{N3}	Specific power shape (W/kg) for stainless steel. Multiplication by AMP gives the total nuclear energy deposition rate. For the point kinetics or power-vs-time options, this is the initial rate. For the quasi-static method, this is updated at reactivity and shape steps.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
EN4	SIMCOM (KOM)	Q_{N4}	Specific power shape (W/kg) for sodium. Multiplication by AMP gives the total nuclear energy deposition rate. For the point kinetics or power-vs-time options, this is the initial rate. For the quasi-static method, this is updated at reactivity and shape steps.
EN5	SIMCOM (KOM)	Q_{N5}	Specific power shape (W/kg) for control material. Multiplication by AMP gives the total nuclear energy deposition rate. For the point kinetics or power-vs-time options, this is the initial rate. For the quasistatic method, this is updated at reactivity and shape steps.
ENCRIT	SIMCOM (KOM)	$\epsilon_{Crt,M}$	Internal energy for material M at the critical point.
ENERGY	SIMCOM (KOM)	Q_{ij}	Total internal energy-production rate for all materials. Used in the URANUS option.
ENORM	SOLVE		The MATSOL error compared with EPSF in Eq. (G-6).
EOSLAS	EOS		Last word of EOS common block.
EPC	SIMCOM (KOM)	ϵ_{G2}	A compressible cell convergence criterion.
EPHASE	SIMCOM (KOM)		Convergence criterion for the iteration involving the mass conservation equations and the EOS in the heat-conduction-limited phase-transition model.
EPI	SIMCOM (KOM)	ϵ_{L2}	An incompressible cell convergence criterion.
EPiR	SIMCOM (KOM)	ϵ_{L1}	The liquid convergence criterion multiplier.
EPS	TIME	ϵ_{G1}	EPS = EPSKAC = convergence precision for compressible cells in the fluid-dynamics pressure iteration.
EPS1	SIMCOM (IA)	ϵ_1	Convergence precision for the amplitude equation solution. Defined as $0.1 * EPSPHI$.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
EPS1	KINET	ϵ_1	Convergence precision for the point kinetics equation solution. Defined as $0.1 * EPSPHI$. Used for the URANUS option.
EPS10	TIME	ϵ_{10}	Maximum number of reactivity steps per power decade.
EPS11	TIME	ϵ_{11}	Maximum-allowed relative change in the total internal energy distribution of the fuel in a reactivity step.
EPS12	TIME	ϵ_{12}	Maximum-allowed relative change in the total internal energy distribution of the stainless steel in a reactivity step.
EPS13	TIME	ϵ_{13}	Maximum-allowed relative change in the total internal energy distribution of the sodium in a reactivity step.
EPS14	TIME	ϵ_{14}	Maximum-allowed relative change in the total density distribution of the fuel in a reactivity step.
EPS15	TIME	ϵ_{15}	Maximum-allowed relative change in the total density distribution of the stainless steel in a reactivity step.
EPS16	TIME	ϵ_{16}	Maximum-allowed relative change in the total density distribution of the sodium in a reactivity step.
EPS17	TIME	ϵ_{17}	Time-step control exponential reduction factor.
EPS18	TIME	ϵ_{18}	Maximum-allowed reactivity change per shape step.
EPS3	KINET	ϵ_3	Point kinetics power-convergence criterion.
EPS4	TIME	ϵ_4	Quasi-static method time-step control reduction factor parameter.
EPS5	TIME	ϵ_5	Maximum-allowed reactivity change per reactivity step.
EPS6	TIME	ϵ_6	Maximum-allowed relative change in the weighted current inner product per shape step.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
EPS7	TIME	ϵ_7	Maximum number of shape steps per power decade.
EPS8	TIME	ϵ_8	Maximum-allowed inverse period change per shape step.
EPS9	TIME	ϵ_9	Maximum-allowed relative spatial tilt in the total flux per shape step.
EPSBKG	SIMCOM (IA)		Convergence criteria for the isotopic background cross-section iteration.
EPSF	SIMCOM (KOM)		Contains the convergence criterion in a call to subroutine MATSOL.
EPSFAC	SIMCOM (IA)		Convergence ratio: rebalance iteration/outer iteration (diffusion option) or inner iteration/outer iteration (transport option).
EPSG	TIME		Convergence precision for the quasistatic method gamma iteration.
EPSK	SIMCOM (KOM)	ϵ_m	The molecular force constant of material M in the vapor phase.
EPSKAC	TIME	ϵ_{G1}	Convergence precision for compressible cells.
EPSL	SIMCOM (KOM)		Matrix inversion convergence criterion for the liquid-component densities.
EPSMIN	SIMCOM (IA)		Minimum convergence precision for inner and rebalance iterations.
EPSO	TIME		Convergence precision for the fission source (outer) iteration.
EPSOK	RXCHA		Temporary variable used in evaluating cells in the fluid dynamics pressure iteration.
EPSP	SIMCOM (KOM)		Matrix inversion convergence criterion for the pressure.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
EPSPHI	SIMCOM (IA)		Convergence precision for the amplitude equation solution.
EPSPHI	KINET		Convergence precision for the point kinetics equation solution. Used for the URANUS option.
EPSPT	SIMCOM (IA)		Convergence precision for point-wise fission source.
EPSR	TIME		Convergence precision for the rebalance iteration (transport option) or the inner iteration (diffusion option).
EPSRBL	SIMCOM (FIA)		Convergence precision for the rebalance iteration (transport option).
EPSV	SIMCOM (KOM)		Matrix inversion convergence criterion for the vapor-component densities.
ERR	RXCHA		Used for the error check in the Newton-Raphson iteration to compute a droplet radius using the Weber number criterion.
ERRFXU	SIMCOM (IA)		Convergence precision in the total fission source for the outer iteration flux shape updates at reactivity steps.
ESFLM	TIME		Cumulative structure-field component specific energy change since the previous postprocessor dump.
ESMOV	TIME		The maximum structure-field component N specific energy change between postprocessor dumps.
EVAPOR	SIMCOM (KOM)		Overall convergence criterion for the heat-conduction-limited phase-transition model.
EXF	FWBGN2		File name for overlay 3,4.
FO	RXCHA		Constant part of the Newton-Raphson function given in Eq. (E-11).

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
F	SOLVE		The solution vector furnished by sub-routine MATSOL.
FACRBL	SIMCOM (IA)		The factor relating the rebalance iteration convergence precision to the outer iteration convergence.
FAIL3	SIMCOM (KOM)		The flag for can-wall failure. FAIL3 = 0,0. radial motion is allowed in the cell; FAIL3 = 1.0 radial motion is not allowed in the cell.
FCORE	SIMCOM (KOM)		Down-force on the core.
FDZ	SIMCOM (KOM)		Fraction of node Δz_j occupied by a movable plug.
FERTWT	SIMCOM (KOM)	W_{Fert}	Reactivity worth distribution of fertile fuel. Used in the URANUS option.
FFAB	RXCHA		The product of FFAB and DTT is the amount of fabricated fuel the XCHANJ subroutine predicts to melt in a time step.
FFLR	SIMCOM (KOM)		The fraction of the heat of fusion that the energy of each structure and solid particle energy component must exceed its solidus energy before failure of the energy component.
FGS1	SIMCOM (KOM)	f_{GS1}	The coefficient in the expression defining the vapor-to-structure friction factor under turbulent flow conditions.
FGS2	SIMCOM (KOM)	F_{GS2}	The Reynolds number exponent in the expression defining the vapor-to-structure friction factor under turbulent flow conditions.
FHEAD	SIMCOM (KOM)		Up-force on the head.
FIA	SIMCOM (IA)		Large array used to contain the neutronic scalar data.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
FIBAR	SIMCOM (KOM)		The floating point equivalent of IBAR.
FIJG	SIMCOM (KOM)	K_{GS}	The vapor-structure momentum exchange function.
FIJL	SIMCOM (KOM)	K_{LS}	The liquid-structure momentum exchange function.
FIRSTA	STAT		First work of STAT common block.
FISSWT	SIMCOM (KOM)	W_{Fiss}	Reactivity worth distribution of fissile fuel. Used in the URANUS option.
FIXL	SIMCOM (KOM)		The floating point frame coordinate for the left edge of plots.
FIXR	SIMCOM (KOM)		The floating point frame coordinate for the right edge of plots.
FIYB	SIMCOM (KOM)		The floating point frame coordinate for the bottom edge of plots.
FIYT	SIMCOM (KOM)		The floating point frame coordinate for the top edge of plots.
FJBAR	SIMCOM (KOM)		The floating point equivalent of JBAR.
FL	SIMCOM (KOM)		The mass-transfer rate of fuel and steel between liquid fuel and steel and solid fuel and steel particles in the liquid field.
FLARGE	CONST		A value used to eliminate machine overflows. Currently set to 10^{20} .
FLLQ5	RXCHA		Initially the energy flux, then the mass flux from fuel particles to liquid fuel in the non-equilibrium melting model.
FLMBUF	KLC		Buffer for film writing and plotting for LTSS option.
FLMOV	TIME		Equivalence array for comparison of post-processor dump input controls with calculated values of the controls.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
FLS1	SIMCOM (KOM)	f_{LS1}	The coefficient in the expression defining the liquid-to-structure friction factor under turbulent flow conditions.
FLS2	SIMCOM (KOM)	f_{LS2}	The Reynolds number exponent in the expression defining the liquid-to-structure friction factor under turbulent flow conditions.
FMAX	RXCHA		The maximum density change of any material in the mass continuity-EOS iteration.
FMED	RXCHA		The term specifying the density change in Eqs. (IV-97) and (IV-100).
FNORM	SIMCOM (IA)		Used in OUTACC to contain the stationary-state eigenvalue or its reciprocal. Used in the matrix solver routines, REBAL, FSOL, and MATSOL, to contain the L_2 norm of the solution vector.
FNRCIN	SIMCOM		The number of real variables in the SIMCOM common block.
FNREOS	EOS		The number of real variables in the EOS common block.
FNRIA	SIMCOM (IA)		The number of real variables in the neutronics SIMCOM common block.
FNRICT	INTCOM (IA)		The number of real variables in the neutronics INTCOM common block.
FNRICT	INTCOM (KOM)		The number of real variables in the hydrodynamics INTCOM common block.
FNRKIN	KINET		The number of real variables in the KINET common block.
FNRPSE	PSEDIT		The number of real variables in the PSEDIT common block.
FNRST1	FWBGN1		The number of real variables in the FWBGN1 common block.
FNRST2	FWBGN2		The number of real variables in the FWBGN2 common block.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
FNRST5	LOCAL		The number of real variables in the LOCAL common block.
FNRSTM	STIM		The number of real variables in the STIM common block.
FNUG	SIMCOM (KOM)	ν_G	The vapor kinematic viscosity used in the KACHINA drag correlation.
FNUG3	SIMCOM (KOM)	$3\nu_G$	Three times the vapor kinematic viscosity used in the KACHINA drag correlation.
FNUL	SIMCOM (KOM)	ν_L	The liquid kinematic viscosity used in the KACHINA drag correlation.
FPLUG	SIMCOM (KOM)		Force on the movable plug.
FPLUGB	SIMCOM (KOM)		Breakaway force before plug can move (input).
FRACHT	RXCHA		Multiplier on external surface areas used to form the structure area available for vaporization-condensation. (See Eq. (III-135)).
FRG	RXCHA	f_G	Single-phase friction factor for vapor-structure drag.
FRGMIN	SIMCOM (KOM)	f_{Gmin}	Input constant representing the minimum single-phase friction factor for vapor-structure drag.
FRL	RXCHA	f_L	Single-phase friction factor for liquid-structure drag.
FRLMIN	SIMCOM (KOM)	f_{Lmin}	Input constant representing the minimum single-phase friction factor for liquid-structure drag.
FS	SIMCOM (KOM)		Mass-transfer rates between the structure and liquid fields.
FSC6	SIMCOM (KOM)		Intragranular fission gas release rate from unrestructured fuel grains for a fuel heating rate of 450 K/s.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
FSC7	SIMCOM (KOM)		A variable containing the constant terms in Eq. (III-253).
FSCAN	RXCHA		Initially energy transfer, then the steel mass transfer rate to or from the can-wall structure.
FSCLD	RXCHA		Initially energy transfer, then the steel mass transfer rate to or from cladding structure.
FSFAB	RXCHA		Initially energy transfer, then the mass flux from fabricated fuel. FSFAB is a negative quantity.
FSHELD	SIMCOM (KOM)		Down force on shield.
FSONE	RXCHA		Mass flux variable in the equilibrium melting model.
FSREF	RXCHA		Initially energy transfer, then the fuel mass flux to or from the can-wall crust.
FST	RXCHA		Melting rate in the equilibrium melting model.
FTEST	RXCHA		Density change convergence criterion for the iteration using the mass conservation equations and the EOS.
FJRSF	SIMCOM (KOM)		The fraction of fabricated fuel that is unrestructured.
FV	SIMCOM (KOM)		Equivalence array for temporary storage of fluid dynamics cell variables for the postprocessor dump.
G	INTCOM (IA)	g	Neutron energy group index.
G	SIMCOM (KOM)	g	Gravitational acceleration constant.
GAM	SIMCOM (KOM)	γ_M	The ratio of the constant-pressure to constant-volume specific heats for material M in the vapor phase.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
GAMCE	SIMCOM (KOM)		Mass-transfer rates for vaporization or condensation.
GAMCOL	KINET	γ_{NA}	Fraction of the initial power generated in sodium.
GAMCON	INTCOM (IA)		Logical variable indicating convergence of the transient flux shape (Gamma) iteration.
GAMCTL	KINET	γ_{Cont}	The fraction of the initial power generated in control material.
GAMFER	KINET	γ_{Fert}	The fraction of the initial power generated in fertile fuel.
GAMFIS	KINET	γ_{Fiss}	The fraction of the initial power generated in fissile fuel.
GAMMA	SIMCOM	γ	Current value of quasistatic method constraint constant.
GAMMAP	SIMCOM (IA)		The value of the quasistatic constraint constant for the previous transient flux shape (Gamma) iteration.
GAMMAX	SIMCOM (IA)		Temporary variable used for the integration of the quasistatic constraint constant.
GAMSS	KINET	ϵ_{SS}	Fraction of the initial power generated in stainless steel.
GDT	SIMCOM (KOM)	$g\Delta t$	The product of the gravitational acceleration constant and the time step.
GENTIM	SIMCOM (IA)	Λ	Current effective prompt neutron generation time.
GENTIN	SIMCOM (IA)	Λ_{ti}	Initial prompt neutron generation time. Used only for the initial state calculation if a transient initial state is desired (ITR = 1).
GENTMS	SIMCOM (IA)	Λ^S	Prompt neutron generation time at the beginning of the transient flux shape time step.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
GGM1	SIMCOM (KOM)	$\gamma_M(\gamma_M - 1)$	The product of GAM and GM1.
GM1	SIMCOM (KOM)	$\gamma_M - 1$	The ratio of the constant-pressure to constant-volume specific heats for material M in the vapor phase minus unity.
GPFLG	SIMCOM (KOM)		Parameter used in modeling to simulate fuel flow in intersubassembly gaps.
GRPBR	RXCHA	$\frac{dp}{dx}$	Pressure gradient used in the Weber number determination of the liquid droplet radii. Defined by Eq. (IV-53).
HA	RXCHA		Used for temperature averages in the multicomponent, phase-transition model.
HALPS	RXCHA	$5.63/(1 - \alpha_S)$	Parameter used in the liquid-liquid heat transfer calculation.
HALPSS	RXCHA	$(1 - \alpha_S)^{-1}$	Reciprocal of the fluid volume fraction for a mesh cell.
HD	RXCHA	H_{LGm}	Product of the surface area and liquid-side heat-transfer coefficient for liquid-field energy component m.
HEADP	SIMCOM (KOM)		Pressure at the center of the head.
HETRAS	RXCHA		Energy transfer rate from liquid to vapor for the case where all liquid of one component vaporizes in a time step.
HFIGJ	RXCHA		Drag heating parameter containing the denominator and q_{KGS} from Eqs. (III-173) thru (III-174).
HFLLF1	RXCHA		Parameter indicating the proportion of the fuel-freezing heat transfer going into crust formation.
HFLLF2	RXCHA		Parameter indicating the proportion of the fuel-freezing heat transfer going into particle formation.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
HFLLS1	RXCHA		Parameter indicating the proportion of the steel-freezing heat transfer going into cladding.
HFLLS2	RXCHA		Parameter indicating the proportion of the steel-freezing heat transfer going into can wall.
HFLLS3	RXCHA		Parameter indicating the proportion of the steel-freezing heat transfer going into particles.
HFUS	EOS	$h_{f,M}$	The heat of fusion of material M.
HGL	RXCHA	h_{GLm}	Vapor-side heat-transfer coefficient for a liquid droplet of energy component m.
HGS	RXCHA	h_{GS}	The vapor-side heat-transfer coefficient for vapor-to-structure heat transfer.
HKI J	RXCHA		Drag heating parameter containing the denominator and q_{KGL} from Eqs. (III-171) thru (III-172).
HL1	RXCHA		Thermal conductivity divided by the particle radius of material one used in liquid-liquid heat transfer.
HL2	RXCHA		Thermal conductivity divided by the particle radius of material two used in liquid-liquid heat transfer.
HLG	RXCHA	h_{LGm}	Liquid-side heat transfer coefficient for a liquid droplet of energy component m.
HLS	RXCHA	h_{LSm}	Heat-transfer coefficient of liquid-field energy component m in contact with a structure.
HLSAL	RXCHA	$\Lambda_m h_{LSm}$	Product used in heat transfer. Obtained from Eqs. (III-126) and (III-129).
HPLTF	FWBGN2		File name for overlays 3,8; 3,9; 3,10; and 3,11.
HS	RXCHA	H_{SGk}	Product of the structure-side heat-transfer coefficient and structure area. Doubly subscripted for future model modification.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
HSO	RXCHA		Same as HS but singly subscripted with the structure index.
HSTAR	SIMCOM (KOM)	$h_{g,m}^*$	The coefficient in the heat of vaporization equation.
HTCCAN	SIMCOM (KOM)	h_{can}^i	The input heat-transfer coefficient for the subassembly can wall in region IREG.
HTCCLD	SIMCOM (KOM)	h_{clad}^i	The input heat-transfer coefficient for cladding in region IREG.
HTCPEL	SIMCOM (KOM)	h_{pell}^i	The input heat-transfer coefficient for fuel pellets in region IREG.
HTCON	RXCHA	h_{Sk}	Heat-transfer coefficient for structure-field energy component m.
HV	RXCHA	H_{GLm}	The corrected product of the vapor-side heat-transfer coefficient and interfacial area per unit volume.
HVO	RXCHA	H_{GLm}^*	The product of the vapor-side heat-transfer coefficient and the interfacial area per unit volume in the absence of phase transitions.
HVP	RXCHA	$H_{GLm}^{\kappa-1}$	The term H_{GLm} on the previous iteration in the heat-transfer limited vaporization/condensation model.
HVP2	RXCHA	$H_{GLm}^{\kappa-2}$	The term $H_{GLm}^{\kappa-1}$ on the previous iteration in the heat-transfer limited vaporization/condensation model.
HVS	RXCHA	H_{GSk}	The corrected product of the structure vapor-side heat-transfer coefficient and the interfacial area per unit volume.
HVSO	RXCHA	H_{GSk}^*	The product of the structure vapor-side heat-transfer coefficient and the interfacial area in the absence of phase transitions.
HVSE	RXCHA	$\hat{H}_{GSk}$	Product of the interfacial area and heat-transfer coefficient used for direct vapor-structure heat transfer. Defined by Eq. (E-19).

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
HVSP	RXCHA	$H_{Gsk}^{\kappa-1}$	The term H_{Gsk} on the previous iteration in the heat-transfer limited vaporization-condensation model.
HVSP2	RXCHA	$H_{Gsk}^{\kappa-2}$	The term $H_{Gsk}^{\kappa-1}$ on the previous iteration in the heat-transfer limited vaporization-condensation model.
HVST	RXCHA		Singly subscripted version of HVS for use in computing mass-transfer modifications.
HX1	RXCHA	$1/r_{pm}$	Inverse droplet radius for liquid energy component m.
HYDF	FWBGN2		File name for overlay 3,0; 3,3; 3,6.
I	SIMCOM (KOM)	i	The radial mesh index.
IA	INTCOM (IA)		Pointer array for neutronics array variables.
IAAROW	INTCOM (KOM)		IAAROW = AAROW; erasable array sufficiently large to contain one row of mesh cells.
IABLT	IXCHA		Index used to specify the type of material on the surface of the pin in the non-equilibrium melting/freezing model.
IAD	LOCAL		Indicator to define the current calculation type: initial adjoint/real flux (0/1).
IALL	FWBGN1		Total length of large-core memory required for the fluid-dynamics calculation.
IASIM	INTCOM (IA)		An array used to contain neutronics integer variables.
IBAR	COMINP	$\bar{i}$	Total number of radial mesh cells.
IBOT	INTCOM (KOM)		Fluid dynamics boundary condition indicator for the bottom boundary.
IBRKXS	UNITS		Logical unit 35, file for the CCCC standard interface file (BRKOXS) of the cross-section shielding factors.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ICAM	UNITS		Film output option.
ICGC	IXCHA		Contains indices for materials that participate in the simple vaporization-condensation model.
ICGL	INTCOM (KOM)		Left boundary of the cover gas region.
ICGR	INTCOM (KOM)		Right boundary of the cover gas region.
ICLC	IXCHA		Contains indices for surfaces in the multicomponent vaporization-condensation model.
ICOMPO	NITIAL		Component number that determines the saturated material to be initially placed in each mesh point set.
ICONV	LOCAL		Fission source (outer) iteration convergence indicator.
ICOS	INTCOM (IA)		Initial flux guess flag.
ICOUNT	IXCHA		Vaporization/condensation index defined by Table E-1.
ICRUST	IXCHA		Index used to specify the type of material on the surface of structure in the non-equilibrium melting/freezing model.
ICW	IXCHA		Index indicating the can-wall set-over condition. Defined in Sec. IV slightly before Eq. (IV-34).
ICWP	IXCHA		Unity if heat was transferred to the present cell from the left; otherwise zero.
IDBFL	PSEDIT		If IDBLT is input as greater than zero, do a debug print after the execution of EXFLUD at the standard print time.
IDFV	SIMCOM (KOM)		A hydrodynamics integer array used to pack data for TAPE36.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IDMP	TIME		Restart dump indicator.
IDTO	INTCOM (KOM)		Index to DTO and DTOC tables.
IDTOP	INTCOM		Index to DTOP and DTOCP tables.
IDUM	INTCOM (KOM)		Flag indicating change of status of a mesh cell from single phase to two phase.
IDUSE	FWBGN1		Problem title.
IEDTSC	PSEDIT		Fluid dynamic time-step control print flag.
IEDXST	PSEDIT		Cross-section print indicator. See input description.
IENEQL	INTCOM (IA)		The subscript translation array from a liquid-field density component to a liquid-field energy component.
IENEQS	INTCOM (KOM)		The subscript translation array from a structure-field density component to a structure-field energy component.
IEOSGR	INTCOM (KOM)		The subscript translation array from a vapor-field density component to a structure-field material component.
IEOSLE	INTCOM (KOM)		The subscript translation array from a liquid-field energy component to a EOS material component.
IEQSSE	INTOCOM (KOM)		The subscript translation array from a structure-field energy component to a EOS material component.
IFAIL	INTCOM (KOM)		The input radial motion indicator for radial mesh cell I.
IFAMF	PSEDIT		Flag to determine if the file size has been specified for TAPE36. Used only for LTSS and CTSS.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IFCL	INTCOM (KOM)		Left boundary for core down-force calculation.
IFCR	INTCOM (KOM)		Right boundary for core down-force calculation.
IFGFAL	ALLSET		Radial motion restraint condition indicator. Defined in the fluid-dynamics input, Card No. 2.
IFINSH	IXCHA		Index indicating the termination condition of the simple vaporization-condensation model.
IFN	S2DIR		An interpolation parameter used in the tabular EOS.
IFNOVY	UNITS		Overlay file name for the CROSS option.
IFNUPD	PSEEDIT		Variable used to control the value of variable LPNEUS (used to force neutronic updates at fluid-dynamic long-print times).
IFSACS	UNITS		SLTAPE1, random file for storing neutronics data.
IFSBCE	UNITS		SLTAPE2, random file for storing fluid-dynamics data.
IFSL	INTCOM (KOM)		Left boundary for the shield down-force calculation.
IFSR	INTCOM (KOM)		Right boundary for the shield down-force calculation.
IFXUDL	INTCOM (IA)		Maximum number of fission source (outer) iterations permitted during reactivity update steps.
IFXULX	INTCOM (IA)		Used to change the value of IFXUDL on restarts.
IFXUPD	INTCOM (IA)		Flag to indicate whether or not a flux shape update is in progress.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IG	INTCOM (IA)		Number of the current neutron energy group.
IGD	INTCOM (IA)		Number of delayed neutron precursor families.
IGEOM	INTCOM (KOM)	§	Horizontal geometry indicator (0 = slab, 1 = cylinder).
IGM	INTCOM (IA)		Number of neutron energy groups.
IGP	INTCOM (IA)		IGM + 1.
IHEAD	PSEDIT		Alphanumeric header for film.
IHV	IXCHA		Index controlling possible values for a droplet's vapor-side heat-transfer coefficient.
IINP	INTCOM (KOM)		Input-type indicator.
IIXMEM	FWBGN1		A pointer relative to the array AA1 pointing to the memory allotted for a memory expansion.
IJ	INTCOM (KOM)		Index for cell (i,j).
IJM	INTCOM (KOM)		Index for cell (i,j-1).
IJMAT	INTCOM (KOM)		Variable indicating the maximum number of cells that can participate in the pressure iteration. Currently IJMAT = 1 000.
IJMAX	INTCOM (KOM)		The total number of cells, IBAR multiplied by JBAR.
IJMAX1	INTCOM (KOM)		IBAR * (JBAR - 1).
IJMAX2	INTCOM (KOM)		IJMAX1 + 1.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IJMAX3	INTCOM (KOM)		IJMAX - 1.
IJP	INTCOM (KOM)		Index for cell (i,j+1).
IK	KINET		Variable index used for storage economy in the point kinetics routine.
IM	COMINP		Total number of neutronics radial mesh cells.
IMAX	SIMCOM (KOM)		Identical to IBAR.
IMAX1	SIMCOM (KOM)		IMAX - 1.
IMAX2	SIMCOM (KOM)		IMAX + 1.
IMF	FWBGN2		File name for overlay (3,5).
IMJMP	INTCOM (IA)		IM * (JM + 1).
IMM1	INTCOM (IA)		IM - 1.
IMMM	INTCOM (IA)		IM * MM.
IMNN	INTCOM (IA)		IM * NN - NN.
IMPJM	INTCOM (IA)		(IM + 1) * JM.
INACS	FWBGN1		First sector of disk used by neutronics in SHUFFL.
INBCS	FWBGN2		First sector of disk used by hydrodynamics in SHUFFL.
INDREG	NITIAL		Region counter.
INPC	COMINP		The array to which all variables in common block COMINP are equivalenced.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IJMAX3	INTCOM (KOM)		IJMAX - 1.
INPRT	PSEDIT		Max (NEULPR, NEUSPR) - flag used to send neutronics input to all devices required for neutronics output.
INTPLT	PSEDIT		Postprocessor plot flag for writing data to TAPE6.
INVPRT	INTCOM (IA)		Inventory print flag (defined by input).
INVREG	INTCOM (IA)		Number of inventory regions.
IOLYCS	INTCOM (IA)		Overlay indicator.
IORCO	ALLSET		Orifice coefficient input indicator. If IORCO > 0, orifice coefficients are input.
IP	INTCOM (IA)		IM + 1.
IP1	COMINP		IBAR + 1.
IP2	COMINP		IBAR + 2.
IPAF	FWBGN1		LCM pointer for the first group of the adjoint flux block.
IPARBL	FWBGN1		LCM pointer for the parameter region data.
IPARBL	FWBGN1		LCM pointer for the parameter region data.
IPASRW	FWBGN1		Controls data swapping in the subroutine SHUFFL.
IPFFAC	FWBGN1		LCM pointer to the first group of shielding factors.
IPFTAB	FWBGN1		LCM pointer to the shielding factor tabulation points.
IPFX	FWBGN1		LCM pointer for the first group of the flux block.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IJMAX3	INTCOM (KOM)		IJMAX - 1.
IPISXS	FWBGN1		LCM pointer to the first group of isotopic cross sections.
IPLFS1	FWBGN1		LCM pointer to the fission source for the previous outer iteration.
IPLFS2	FWBGN1		LCM pointer to the fission source for the second preceding outer iteration.
IPLKC	FWBGN1		LCM pointer to the leakage reactivity component. Used only if RHOMAP is defined.
IPLKL	FWBGN1		LCM pointer to the leakage reactivity component of the previous shape step. Used only if RHOMAP is defined.
IPNO	FWBGN1		Problem code number.
IPOWER	KINET		Energy source option indicator, no source/ reactivity/power (-1/0/1).
IPPC	FWBGN1		LCM pointer for the first delayed neutron precursor family concentration.
IPPF	FWBGN1		LCM pointer for the first group of the flux at the previous shape step.
IPPK	FWBGN1		LCM pointer for the amplitude equation parameters of the first reactivity step during the flux-shape time step.
IPQ	FWBGN1		LCM pointer for the group source exclusive of the in-group scattering (transport option only).
IPREAN	KINET		The number of entries in the reactivity-or power-vs-time table.
IPSTEP	TIME		Reactivity time-step number relative to the beginning of the flux shape time step.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IJMAX3	INTCOM (KOM)		IJMAX - 1.
IPTMAX	INTCOM (KOM)		Radial location of the peak pressure on the head.
IPXSCT	FWBGN1		LCM pointer for the first group of the spatial macroscopic cross-section array.
IQUASI	TIME		Flag to override the quasistatic time step controls, as defined by input.
IRBM	TIME		Maximum number of rebalance iterations per group.
IRCOVY	UNITS		Overlay recall indicator for CDC machines (0/6HRECALL = no reloading of overlay when in core/reload overlay).
IREBAL	INTCOM (IA)		Total number of rebalance iterations per inner iteration.
IREBLT	INTCOM (IA)		Total number of rebalance iterations of a given group of a given outer iteration.
IRHOCK	IXCHA		Iteration counter for the iteration defining the gas constant in the quasi-equilibrium vaporization-condensation algorithm.
IRHOIT	IXCHA		Iteration counter for the mass-continuity/EOS iteration in the simple phase transition model.
IRHOMP	INTCOM (IA)		Input variable indicating the type of postprocessor (TAPE6) output desired.
IRIGHT	INTCOM (KOM)		Input boundary condition indicator for the right-hand boundary.
IRSTEP	TIME		Number of reactivity time steps relative to the beginning of the problem.
IRTEDF	STIM		Steady-state/transient (0/1) indicator for execution time edit.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IJMAX3	INTCOM (KOM)		IJMAX - 1.
ISABRT	PSEDIT		Abort flag for input errors. Has been disengaged.
ISAT	INTCOM (KOM)		An indicator to check if input liquid-field component temperatures exceed saturation temperatures when saturation conditions are desired.
ISCFIA	FWBGN1		Number of sectors of disk space needed for the neutronics SIMCOM common block.
ISCFKM	FWBGN1		Number of sectors of disk space needed for the hydrodynamics SIMCOM common block.
ISCIA	FWBGN1		Number of sectors of disk space needed for the neutronics LCM common block.
ISCIIA	FWBGN1		Number of sectors of disk space needed for the neutronics INTCOM common block.
ISCIKM	FWBGN1		Number of sectors of disk space needed for the hydrodynamics INTCOM common block.
ISCKOM	FWBGN1		Number of sectors of disk space needed for the fluid dynamics LCM common block.
ISCS	IXCHA		Material index for surfaces in the multi-component phase transition model.
ISHLD	INTCOM (IA)		Flag to defeat cross-section shielding. Normal shielding/no shielding = 0/1.
ISIGOD	INTCOM (IA)		Number of background points used for Lagrange interpolation of shielding factors.
ISK36	PSEDIT		Fluid dynamics postprocessor indicator. (Postprocessor dumps/no postprocessor dumps = 0/1).
ISN	INTCOM (IA)		Order of S_n approximation.
ISNCON	UNITS		Logical unit 32, file for input of S_n constants.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ISNT	LOCAL		S_n library request indicator.
ISO	INTCOM (KOM)		Right-hand boundary of plug.
ISOTXS	UNITS		Logical unit 34, file for input of the multigroup cross-sections.
ISPR	PSEDT		Fluid dynamics short print selection flag.
ISS	IXCHA		DO LOOP index over surfaces in the multi-component phase-transition model.
ISTEP	TIME		Indicates the time-step control that determines the fluid dynamics time step.
ISTRHT	IXCHA		Flag on direct structure-vapor heat transfer. Direct heat transfer/no direct heat transfer = 1/0.
ISURF	IXCHA		Flag to distinguish if a surface is a solid or a liquid in the multicomponent phase-transition model.
IT	COMINP		IT is the total number of neutronics radial mesh cells. Equivalenced to IM.
IT	SOLV		Iteration counter for subroutine MATSOL.
ITEMOD	INTCOM (IA)		Number of temperature points used for Lagrange interpolation of shielding factors.
ITIS	STIM		An array equivalenced to the first variable of the STIM common block to allow treating the whole common block as an array.
ITGAM	LOCAL		The current number of quasi-static method gamma iterations.
ITGAMM	INTCOM (IA)		The maximum number of quasi-static method gamma iterations permitted.
ITIMF	TIME		An index describing the type of time limit being used by SIMMER-II (1/2/3 - from dump/from input/from execute line).

TABLE J-I (CONT)

<u>FORTAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ITH	LOCAL		Neutronics initialization type indicator (steady state/transient = 0/1).
ITJT	INTCOM (IA)		IT * JT.
ITLMBG	INTCOM (IA)		Maximum number of iterations over the background cross section permitted.
ITLMIN	LOCAL		Maximum number of inner (scattering-source) iterations. Used in the transport theory option.
ITLMOU	LOCAL		Maximum number of outer (fission-source) iterations permitted.
ITNR	IXCHA		A counter for the multicomponent phase calculation.
ITOP	INTCOM (IA)		Boundary condition type indicator for the top boundary.
ITOPT	TIME		Optimal number of fluid dynamics pressure iterations per time step.
ITP	COMINP		IT + 1.
ITP10	INTCOM (KOM)		The number of consecutive cycles exceeding the maximum-allowed pressure iterations.
ITK	INTCOM (IA)		Input neutronics transient initialization flag.
ITRO10	INTCOM (KOM)		The number of consecutive cycles exceeding the maximum allowed vapor iterations.
ITS	FWBGN2		Restart indicator.
ITSP	INTCOM (KOM)		Flag in the simple phase-transition model defined in Table E-I.
IVIS	INTCOM (KOM)		Input viscous shear indicator.
IVPALL	IXCHA		If set to 1, IVPALL indicates all the liquid of a given component is vaporizing in a time step.

TABLE J-I (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
IVPALP	IXCHA		Index to limit oscillations in the value of IVPALL.
IVI	TIME		Array containing the radial indices of the cells limiting the DTL time-step controls.
IW12	TIME		Variable to temporarily store IVI(12) in subroutine HYDRO.
IW4	TIME		Variable to temporarily store IVI(4) in subroutine HYDRO.
IW7	TIME		Variable to temporarily store IVI(7) in subroutine HYDRO.
IWTF	LOCAL		Weighting function used for the amplitude equation parameter integrations.
IXL	INTCOM (KOM)		Integer frame coordinate for left edge of plots.
IXMEM	FWBGN1		A pointer relative to AA1(1) pointing to the memory received by a memory expansion.
IXR	INTCOM (KOM)		Integer frame coordinate for right edge of plots.
IXSFN	UNITS		File name for the cross-section random-access file.
IXSPTS	INTCOM (IA)		Isotope number density initial print flag.
IXSPTT	INTCOM (IA)		Isotope number density transient print flag.
IY	RTBLK2		Index of the Y vector in subroutine RATFN2 (tabular EOS).
IYB	INTCOM (KOM)		Integer frame coordinate for the bottom edge of plots.
IYT	INTCOM (KOM)		Integer frame coordinate for the top edge of plots.
J	INTCOM (KOM)		Axial mesh index.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
J2	INTCOM (KOM)		First word address of the row of fluid dynamics mesh cells for which the calculation is currently in progress.
J3	INTCOM (KOM)		First word address of the row of fluid dynamics mesh cells above the row for which the calculation is currently in progress.
JBAR	COMINP	$\bar{J}$	Total number of axial mesh cells.
JCGB	INTCOM (KOM)		Bottom boundary of the cover gas region.
JCGT	INTCOM (KOM)		Top boundary of the cover gas region.
JCGV	PSEDIT		The number of cells from the top of the problem to include in the cover gas calculation for the long print.
JFC	INTCOM (KOM)		Node row for calculating core downforce.
JFS	INTCOM (KOM)		Node row for calculating the annular shield downforce.
JM	COMINP		Total number of neutronics axial mesh cells.
JMAX	INTCOM (KOM)		Identical to JBAR.
JMAX1	INTCOM (KOM)		JMAX - 1.
JMM	INTCOM (IA)		JT * MM.
JP	INTCOM (IA)		JM + 1.
JP1	COMINP		JBAR + 1.
JP2	COMINP		JBAR + 2.

TABLE J-1 (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
JPLUGI	INTCOM (KOM)		Axial index for the location of the plug.
JPLUGS	INTCOM (KOM)		Index to track crossing of node boundaries. If JPLUGS = 0, no boundaries are crossed. If JPLUGS = 1, an upward boundary is crossed. If JPLUGS = 2, a downward boundary is crossed.
JRIGID	INTCOM (KOM)		The number of mesh cells up the right boundary that are to have a rigid, free- slip boundary condition.
JSO	INTCOM (KOM)		Bottom boundary of plug.
JSOT	INTCOM (KOM)		Top boundary of plug.
JT	COMINP		Total number of neutronics axial mesh cells.
JTOP	INTCOM (KOM)		JTOP = JP1 = JBAR + 1 if top is rigid; JTOP = JP2 = JBAR + 2 if top is outflow; JTOP controls the velocity initialization in the implicit calculation.
JTP	COMINP		JT + 1.
JVJ	TIME		Array containing the axial indices of the cells limiting the DTL time-step controls.
JW12	TIME		Variable to temporarily store JVJ(12) in subroutine HYDRO.
JW4	TIME		Variable to temporarily store JVJ(4) in subroutine HYDRO.
JW7	TIME		Variable to temporarily store JVJ(7) in subroutine HYDRO.
JX1	INTCOM (KOM)		First word address of the second fluid dy- namics mesh cell in row J = 1, set by LOOP as an index.
JX2	INTCOM (KOM)		First word address of the second fluid dy- namics mesh cell in row J = 2, set by LOOP as an index.

TABLE J-1 (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
JX3	INTCOM (KOM)		First word address of the second fluid dynamics mesh cell in row $J = 3$, set by LOOP as an index.
KALL	KINET		LCM pointer for the point kinetics mesh-cell-dependent data.
KCHAN	IXCHA		Structure configuration model variable indicating the sequence of treating re-frozen fuel and control material.
KCHPS	IXCHA		Structure configuration model variable used in combination with KCHAN to layer refrozen fuel and control material.
KECR	KINET		LCM pointer for reading the next row of point kinetics values into SCM.
KECW	KINET		LCM pointer for writing the next row of point kinetics values to LCM.
KIJ	SIMCOM (KOM)		Liquid-vapor momentum exchange function.
KIJDTP	RXCHA		Liquid-vapor momentum exchange function on the previous time step.
KINLAS	KINET		Last word of KINET common block.
KNTIN	INTCOM (IA)		Inner iteration counter.
KOMLAS	SIMCOM (KOM)		Last word of the INTCOM(KOM) common block.
KPOW	KINET		Point kinetics equation solution indicator for subroutine HTSPK 1/2/3 = initial pass/transient pass/second transient pass.
KSTEP	TIME		Flux-shape time-step number.
KSTLIM	TIME		Number of initial shape steps required before DELTAT may be increased.
KTDMF	TIME		Control index used for location in the TDMP and DTDMP arrays in selecting times for restart dumps.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
L36NAM	UNITS		File name of fluid dynamics postprocessor data file.
L3DM	PSEEDIT		Number of movie frames repeated for each edit time step.
L3DMVR	PSEEDIT		Flag for selecting fluid dynamics state variables to be used in perspective movies.
L3DS	PSEEDIT		Flag for the fluid dynamics plotting option.
LA1	INTCOM (IA)		The pointer to the neutronics variable array containing $2\pi (r_{i+1/2} + r_{i-1/2})$.
LA2	INTCOM (IA)		The pointer to the neutronics variable array containing $2\pi (r_{i+1/2} - r_{i-1/2})$.
LA3	INTCOM (IA)		The pointer to the neutronics variable array containing $\pi (r_{i+1/2}^2 - r_{i-1/2}^2)$.
LA4	INTCOM (IA)		The pointer to the neutronics variable array containing $2\pi r_{i-1/2}$.
LA5	INTCOM (IA)		The pointer to the neutronics variable array containing $\pi (r_{i+1/2}^2 - r_{i-1/2}^2)$.
LA6	INTCOM (IA)		The pointer to the neutronics variable array containing $(r_{i+1} - r_i)$.
LAASC	FWBGN2		The length of the AASC array in INTCOM(KOM).
LAB	INTCOM (IA)		The pointer to the neutronics variable array containing the removal rate used in the rebalance equations.
LAFLUX	INTCOM (IA)		The pointer to the neutronics variable array containing the adjoint flux. This shares common with the A(LQ) array.
LAH	INTCOM (IA)		The pointer to the neutronics variable array containing a weighted flux amplitude integral for the spatial precursor concentration solution.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LAL1	INTCOM (IA)		The pointer to the neutronics variable array containing $\alpha_{m+1/2}/w_m$.
LAL2	INTCOM (IA)		The pointer to the neutronics variable array containing $\alpha_{m-1/2}/w_m$.
LALF	INTCOM (IA)		The pointer to the neutronics variable containing the flux because of curvature streaming.
LANUE	FWBGN1		The length of neutronics arrays A and FIA.
LAQQ	INTCOM (KOM)		The pointer to the neutronics variable containing the source terms for the transport theory inner iteration. Shares common with the A(LF) array. Also used as temporary storage by subroutines OUTER, REBAL, INNET, FSOL, INNED, and POWCAL.
LAST	FWBGN1		Length of neutronics common array A used by the problem.
LASTAT	STAT		Last word of the STAT common block.
LASTEC	FWBGN1		Length of LCM used by the space-dependent neutronics.
LASITC	INTCOM (IA)		The last word of INTCOM common block of the neutronics part of the code.
LASITK	INTCOM (KOM)		The last word of INTCOM common block of the hydrodynamics part of the code.
LASTS	STIM		Last word of the STIM common block.
LASWOR	SIMCOM (KOM)		Last word of the SIMCOM common block used by the fluid dynamics.
LASWOR	SIMCOM (IA)		The last word of the SIMCOM common block of the neutronics part of the code.
LB1	INTCOM (IA)		The pointer to the neutronics variable array containing $1/(z_{j+1/2} - z_{j-1/2})$.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LBATA	INTCOM (IA)		The pointer to the neutronics variable array containing the effective delayed neutron fraction divided by the effective generation time. Shares common with A(LF).
LBETA	INTCOM (IA)	β_i^P	The pointer to the neutronics variable array containing the adjusted physical delayed neutron fraction.
LBETAO	INTCOM (IA)	β_i^P	The pointer to the neutronics variable array containing the input physical delayed neutron fraction. Shares common with A(LBETVA).
LBETCO	INTCOM (IA)		The pointer to the neutronics variable array containing the parabola coefficients for the effective delayed neutron fraction divided by the generation time.
LBETIN	INTCOM	β_i	The pointer to the neutronics variable array containing the input effective delayed neutron fraction.
LBETVA	INTCOM (IA)		The pointer for the neutronics variable array containing tabular values of effective delayed neutron fraction divided by the generation time.
LBH	INTCOM (IA)		The pointer to the neutronics variable array containing a weighted flux amplitude integral for the spatial precursor concentration solution.
LBINT	INTCOM (IA)		The pointer to a neutronics variable array containing temporary storage for the amplitude equation solution. Shares common with A(LF).
LBR1	INTCOM (IA)		The pointer to the neutronics variable array containing the right cell boundary flux (in-down and out-down). Used for the transport-theory option.
LBR2	INTCOM (IA)		The pointer to the neutronics variable array containing the right cell boundary flux (in-up and out-up). Used for the transport-theory option.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LBT1	INTCOM (IA)		The pointer to the neutronics variable array containing the top cell boundary flux (in-down and out-up). Used for the transport-theory option.
LBT2	INTCOM (IA)		The pointer to the neutronics variable array containing the top cell boundary flux (in-down and in-up). Used for the transport-theory option.
LC	INTCOM (IA)		The pointer to the neutronics variable array containing the microscopic isotopic unshielded sections for a single energy group.
LCBURN	INTCOM (IA)		The pointer to the neutronics variable array containing temporary storage for the amplitude equation solution. Shares common with A(LF).
LCD	INTCOM (IA)		The pointer to the neutronics variable array containing material component densities.
LCE	INTCOM (IA)	η_m	The pointer to the neutronics variable array containing the axial direction cosines. Used for the transport-theory option.
LCELPT	INTCOM (IA)		The pointer to the neutronics variable array containing the corner neutronics cells to the isotopic inventory regions.
LCHI	INTCOM (IA)	x_g	The pointer to the neutronics variable array containing the prompt fission spectrum.
LCM	INTCOM (IA)		The pointer to the neutronics variable array containing the radial direction cosine. Used for the transport option.
LCMNUE	FWBGN1		The length of LCM used by the space neutronics.
LCMX	S2DIR		Length of array TBLS, which stores the tables for the SESAME equation of state.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LCPDT	INTCOM (IA)		The pointer to the neutronics variable array containing the densities and temperatures for each neutronics material.
LCTOT	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise macroscopic cross-sections.
LDBETA	INTCOM (IA)		The pointer to the neutronics variable array containing the change in the effective delayed neutron fraction over a flux-shape time step.
LDC	INTCOM (IA)		The pointer to the neutronics variable array containing the assignment of neutronics mixtures to spatial cross-section regions.
LDENST	FWBGN1		Length of density/temperature array in large memory. Used only if the URANUS option is off.
LDEQ	INTCOM (IA)		The pointer to a neutronics variable array containing temporary storage for amplitude equation solution. Shares common with A(LF).
LDFISS	INTCOM (IA)		The pointer to the neutronics variable array containing the weighted delayed neutron fission source integral for reactivity calculation.
LDISPL	FWBGN1		Length of the displacement table in LCM. (Note: This feature has not been implemented.)
LDMPOU	FWBGN2		Length of SIMMER-II restart dump. Used only in the LTSS version.
LEBETA	INTCOM (IA)		The pointer to the neutronics variable array containing the effective delayed neutron fractions.
LECDEN	FWBGN1		Large-memory pointer for the density/temperature array. Used only if the URANUS option is off.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LECDIS	FWBGN1		Large-memory pointer for the displacement array. (Note: This feature has not been implemented.)
LECELL	FWBGN1		Large-memory pointer for storage region swapped between the fluid dynamics variables and neutronics variables.
LECENT	FWBGN1		Large-memory pointer for the ENERGY array; used only if the URANUS option is off.
LEN	INTCOM (IA)		The pointer to a neutronics variable array containing temporary storage for the amplitude equation solution. Shares common with A(LF).
LEN36	PSEDIT		Length of the postprocessor file TAPE36.
LENCIA	FWBGN2		Length of the array IA in INTCOM(IA).
LENERG	FWBGN1		Length of the ENERGY array in large core.
LENFIA	FWBGN2 (IA)		Length of the FIA array in INTCOM common block.
LENHSM	FWBGN1		Length of the fluid dynamics INTCOM common block.
LENSCT	FWBGN1		Number of words in a disk sector.
LENTAP	FWBGN1		Maximum length of TAPE36, the postprocessor file. Currently set at 13 000 000 decimal words.
LETAIN	INTCOM (IA)		The pointer to a neutronics variable array containing the initial integral precursor concentrations for transient initialization of the time-dependent neutronics solution.
LF	INTCOM (IA)		The pointer to the neutronics variable array containing fine mesh rebalance factors for use by subroutines REBAL and INNET. Also used for temporary storage by subroutines SHLDXS and INPROD.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Commo . Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LFD	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise downward partial currents used for rebalance factors. Also used for temporary storage by subroutine FSOL.
LFFBUF	INTCOM (IA)		The pointer to the neutronics variable array used as a buffer for input of isotopic shielding factors.
LFFCEL	INTCOM (IA)		The pointer to the neutronics variable array containing interpolated shielding factors for a mesh cell.
LFFF	INTCOM (IA)		The pointer to the neutronics variable array containing cross-section shielding factors for an energy group.
LFISA	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise fission source used by subroutines OUTER, INPROD, and PKDRIV.
LFISA	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise fission source used by subroutines OUTER, INPROD, and PKDRIV.
LFL	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise scalar flux for an energy group. Used by subroutines OUTER, INNET, FSOL, INNED, POWCAL, INPROD, and PKDRIV. Also used as temporary storage by subroutines NPUT15, GRIND, SHLDXS, OUTER, OUTACC, POWCAL, INPROD, PKDRIV, and PCINT.
LFL	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise leftward partial currents used for rebalance factors. Also used for temporary storage by subroutine FSOL.
LFR	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise rightward partial currents used for rebalance factors. Also used for temporary storage by subroutine FSOL.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LFU	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise upward partial currents used for rebalance factors. Also used for temporary storage by subroutine FSOL.
LGA	INTCOM (IA)		The pointer to a neutronics variable array used for solution of neutronics matrix equations.
LHA	INTCOM (IA)		The pointer to a neutronics variable array used in solution of neutronics matrix equations.
LIANUE	FWBGN1		The number of words in the neutronics INTCOM common block.
LIFFB	INTCOM (IA)		The number of distinct isotopic cross-section shielding factor types. (Equal to 3 for transport theory and 4 for diffusion theory).
LIFFC	INTCOM (IA)		The table position of the isotopic capture cross-section shielding factor.
LIFFE	INTCOM (IA)		The table position of the isotopic elastic cross-section shielding factor.
LIFFF	INTCOM (IA)		The table position of the isotopic fission cross-section shielding factor.
LIFFT	INTCOM (IA)		The table position of the isotopic transport cross-section shielding factor.
LINE	PSEDIT		Long print line selector. Input in fluid-dynamics input Card No. 4.
LINVRG	INTCOM (IA)		The pointer to the neutronics variable array containing the boundary locations of the isotopic inventory regions.
LIPSTP	TIME		The input maximum number of reactivity time steps per flux-shape step.
LISCAP	INTCOM (IA)		The table position of the isotopic capture cross section.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LISCDN	INTCOM (IA)		The pointer to the neutronics variable array containing the isotopic number densities for each material for a mesh cell.
LISDEN	INTCOM (IA)		The pointer to the neutronics variable array containing the isotopic number densities summed over all neutronic materials for a mesh cell.
LISDWN	INTCOM (IA)		The table position of the first isotopic down scatter cross section.
LISEI1	INTCOM (IA)		The table position of the P1 component of the isotopic elastic within-group scattering cross section.
LISELI	INTCOM (IA)		The table position of the isotopic elastic within-group scattering cross section.
LISELO	INTCOM (IA)		The table position of the isotopic elastic down scatter cross section.
LISE01	INTCOM (IA)		The table position of the P1 component of the elastic down scatter cross section.
LISFIS	INTCOM (IA)		The table position of the isotopic fission cross section.
LISII1	INTCOM (IA)		The table position of the P1 component of the isotopic inelastic within-group scattering cross section.
LISING	INTCOM (IA)		The table position of the combined isotopic inelastic ingroup and (N,2N) ingroup scattering cross sections. (The (N,2N) reaction is additive).
LISI01	INTCOM (IA)		The table position of the first P1 component of the isotopic inelastic down scatter cross sections.
LISLOC	INTCOM (IA)		The pointer to the neutronic variable array containing the locations of desired isotopes on the ISOTXS input cross-section file.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LISMAX	INTCOM (IA)		The length of the isotopic cross-section tables.
LISMSC	INTCOM (IA)		The table position of the isotopic miscellaneous cross section.
LISNSF	INTCOM (IA)		The table position of the product of the isotopic fission cross section and the number of emitted neutrons.
LISOFF	INTCOM (IA)		The pointer to the neutronics variable array containing input shielding factors from BRKOXS file.
LISONM	INTCOM (IA)		The pointer to the neutronics variable array containing isotope names.
LISOXS	INTCOM (IA)		The pointer to the neutronics variable array containing input isotopic cross sections from the ISOTXS file.
LISREM	INTCOM (IA)		The pointer to the neutronics variable array containing the isotopic removal cross section.
LISSWP	INTCOM (IA)		The pointer to the neutronics variable array containing flags indicating the necessity to reverse the ordering of input background shielding factors.
LISSWT	INTCOM (IA)		The pointer to the neutronics variable array containing flags indicating the necessity to reverse the ordering of input temperature shielding factors.
LISTMP	INTCOM (IA)		The pointer to the neutronics variable array containing the averaged isotopic temperatures for a mesh cell.
LISTRN	INTCOM (IA)		The table position of the isotopic transport cross section.
LITKOM	FWBGN1		The number of words in the hydrodynamics common block INTCOM.
LLALCM	FWBGN1		The maximum number of words in LCM needed by SIMMER.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LLAM	INTCOM (IA)		The pointer to the neutronics variable array containing the delayed neutron precursor decay constants.
LLC	INTCOM (IA)		The pointer to the neutronics variable array containing the current delayed neutron precursor concentration integrals.
LLCM	FWBGN1		The last word of LCM before any memory expansion is done.
LLCO	INTCOM (IA)		The pointer to the neutronics variable array containing the weighted delayed neutron precursor integrals from a previous time step.
LLFFGP	INTCOM (IA)		The pointer to the neutronics variable array containing secondary pointers to each block of isotopic shielding factors for an energy group.
LLIQ	IXCHA		Used as a subscript for the droplet diameter in the multicomponent phase-transition model.
LLMN	INTCOM (IA)		The pointer to a neutronics variable array containing secondary pointers to identify the beginning of isotopic data for each material in any of several arrays relating to the neutronic materials description.
LLOCA	INTCOM (IA)		The pointer to the neutronics variable array containing the record addresses of individual isotopes on the input ISOTXS cross-section file.
LLSIGT	INTCOM (IA)		The pointer to the neutronics variable array containing secondary pointers to the block of tabulation points for the background dependence of shielding factors for each isotope.
LLSTEC	FWBGN1		Pointer to the first location of large core not in either the fluid dynamics or neutronics large core data.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LLTEMT	INTCOM (IA)		The pointer to the neutronics variable array containing secondary pointers to the block of tabulation points for the temperature dependence of shielding factors for each isotope.
LMC	INTCOM (IA)		The pointer to the neutronics variable array containing identifiers for the isotopes composing each neutronics material.
LMD	INTCOM (IA)		The pointer to the neutronics variable array containing number densities for each isotope composing each neutronics material.
LMTBLK	INTCOM (IA)		The total number of basic cross-section types (capture, fission, etc.) in the LCM block of cross sections for each spatial mesh cell.
LMTDWN	INTCOM (IA)		The table position of the beginning of the LCM block of downscatter macroscopic cross sections for all spatial mesh cells.
LMTFCR	INTCOM (IA)		The table position of the LCM block of radial finite-difference coefficients for the diffusion equation.
LMTFCU	INTCOM (IA)		The table position of the LCM block of axial finite-difference coefficients for the diffusion equation.
LMTFDI	INTCOM (IA)		The table position of the LCM block of central finite-difference coefficients for the diffusion equation.
LMTIGS	INTCOM (IA)		The table position of the LCM block of within-group scattering macroscopic cross sections for all spatial mesh cells.
LMTMAX	INTCOM (IA)		The total number of cross-section types in the LCM block of macroscopic cross sections for all spatial mesh cells.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LMNSF	INTCOM (IA)		The table position of the LCM block of the product of the number of fission neutrons and the fission macroscopic cross section for all spatial mesh cells.
LMTREM	INTCOM (IA)		The table position of the LCM block of the removal (for the diffusion theory option) or the total (for the transport theory option) macroscopic cross section for all spatial mesh cells.
LMTTRN	INTCOM (IA)		The table position of the large core memory block of the transport macroscopic cross-section for all spatial mesh cells.
LNFFGP	INTCOM (IA)		The pointer to the neutronics variable array containing the number of isotopic shielding factors for an energy group.
LNHCIN	COMINP		The number of words containing Hollerith data in the COMINP common block.
LNHEOS	EOS		The number of words containing Hollerith data in the EOS common block.
LNHITC	INTCOM (IA)		The number of words containing Hollerith data in the INTCOM common block for neutronics.
LNHITK	INTCOM (KOM)		The number of words containing Hollerith data in the INTCOM common block for fluid dynamics.
LNHKIN	KINET		The number of words containing Hollerith data in the KINET common block.
LNHKOM	SIMCOM (KOM)		The number of words containing Hollerith data in the SIMCOM common block for fluid dynamics.
LNHPSE	PSEDIT		The number of words containing Hollerith data in the PSEDIT common block.
LNHST1	FWBGN1		The number of words containing Hollerith data in the FWBGN1 common block.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LNHST2	FWBGN2		The number of words containing Hollerith data in the FWBGN2 common block.
LNHST5	LOCAL		The number of words containing Hollerith data in the LOCAL common block.
LNHSTM	STIM		The number of words containing Hollerith data in the STIM common block.
LNHTIM	TIME		The number of words containing Hollerith data in the TIME common block.
LNICIN	COMINP		The number of words containing integer data in the COMINP common block.
LNIEOS	EOS		The number of words containing integer data in the EOS common block.
LNIITC	INTCOM (IA)		The number of words containing integer data in the INTCOM common block for neu- tronics.
LNIITK	INTCOM (KOM)		The number of words containing integer data in the INTCOM common block for fluid dynamics.
LNIKIN	KINET		The number of words containing integer data in the KINET common block.
LNIKOM	SIMCOM (KOM)		The number of words containing integer data in the SIMCOM common block for fluid dynamics.
LNIPSE	PSEDIT		The number of words containing integer data in the PSEDIT common block.
LNIST1	FWBGN1		The number of words containing integer data in the FWBGN1 common block.
LNIST2	FWBGN2		The number of words containing integer data in the FWBGN2 common block.
LNIST5	LOCAL		The number of words containing integer data in the LOCAL common block.
LNISTM	STIM		The number of words containing integer data in the STIM common block.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LNISTP	INTCOM (IA)		The number of words containing integer data in the INTCOM common block for neutronics.
LNITIM	TIME		The number of words containing integer data in the TIME common block.
LMN	INTCOM (IA)		The pointer to the neutronics variable array containing the number of variables for each material in any of several arrays relating to the neutronic materials description.
LNSIGT	INTCOM (IA)		The pointer to the neutronics variable array containing the size of the block of tabulation points for the background dependence of shielding factors for each isotope.
LNTEMT	INTCOM (IA)		The pointer to the neutronics variable array containing the size of the block of tabulation points for the temperature dependence of shielding factors for each isotope.
LO	INTCOM (IA)		The pointer to the neutronics variable array containing the time-step expansion coefficients used to solve the amplitude equation. Shares common with A(LF).
LOCORF	INTCOM (KOM)		The input axial orifice location. The input value is incremented by 1 so that it is consistent with internal SIMMER-II numbering.
LOCRRS	INTCOM (KOM)		LCM pointer to allow LCM space for density and energy array storage in IMFLUD.
LOMEG	INTCOM (IA)		The pointer to the neutronics variable array containing overrelaxation parameters. Used to solve the rebalance equations (for the transport theory option) or cell-wise flux (for the diffusion theory option).
LPARBL	FWBGN1		Total length of the fluid dynamics parameter region block.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LPC	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise delayed neutron precursor concentrations.
LPCINT	INTCOM (IA)		The pointer to the neutronics variable array containing the integral of the spatial delayed neutron precursor concentrations.
LPCNRM	INTCOM (IA)		The pointer to the neutronics variable array containing factors used to renormalize the delayed neutron source during transient initialization of the time dependent neutronics solution.
LPCS	INTCOM (IA)		The pointer to the neutronics variable array containing the delayed neutron precursor concentration integrals from the previous flux-shape time step.
LPFISS	INTCOM (IA)		The pointer to the neutronics variable array containing the fission source from the previous flux-shape time step. Shares common with A(LAB).
LPFLUX	INTCOM (IA)		The pointer to the neutronics variable array containing the flux from the previous flux-shape time step. Shares common with A(LAB).
LPNEUS	TIME		Variable forcing neutronic updates at fluid-dynamic long-print times.
LPPK	INTCOM (IA)		First word of the amplitude equation parameter block.
LPR	PSEDIT		Fluid dynamics output selection variable.
LPRS	PSEDIT		Internal value of LPR, used for input edit.
LPRSTP	PSEDIT		Long print flag. Print/no print = 1/0.
LPS	INTCOM (IA)		The pointer to the neutronics variable array containing the input delayed neutron emission spectra.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LQ	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise source (except for the within - group scattering contribution) for the space-angle sweep portion of the transport inner iteration.
LQQ	INTCOM (IA)		The pointer to the neutronics variable array containing the cell-wise fine-mesh re-balance source used for the transport theory solution.
LR36	PSEDIT		Record length for the fluid dynamics post-processing file.
LREAT	INTCOM (IA)		The pointer to the neutronics variable array containing the parabola coefficients used to extrapolate the reactivity to advanced times.
LRXSPT	INTCOM (IA)		The pointer to the neutronics variable array containing the assignment of region indicators to each mesh cell.
LSBETT	INTCOM (IA)		The pointer to the neutronics variable array containing the effective delayed neutron fraction from the previous flux-shape time step.
LSCM	FWBGN1		The last word of SCM before any memory expansion is done.
LSIGO	INTCOM (IA)		The pointer to the neutronic variable array containing the last-iterate background cross section for each isotope in the current mesh cell.
LSIGON	INTCOM (IA)		The pointer to the neutronic variable array containing the updated background cross section for each isotope in the current mesh cell.
LSIGP	INTCOM (IA)		The pointer to the neutronic variable array containing the tabulation points for the background dependence of shielding factors.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LSIGRW	INTCOM (IA)		The pointer to the neutronic variable array containing the cell-wise macroscopic elastic down scatter cross section from the next higher energy group.
LSIGT	INTCOM (IA)		The pointer to the neutronic variable array containing the tabulation points for the temperature dependence of shielding factors.
LSMREA	INTCOM (IA)		The pointer to a neutronics variable array containing temporary storage for the amplitude equation solution. Equivalenced to LSMRES.
LSMRES	INTCOM (IA)		Equivalenced to LSMREA.
LSUMB	INTCOM (IA)		The pointer to a neutronics variable array containing temporary storage for the amplitude equation solution. Shares common with A(LF).
LSOU	INTCOM (IA)		The pointer to the neutronics variable array containing the total cell-wise source for the space-angle sweep portion of the transport inner iteration. Shares common with A(LQ).
LT	INTCOM (IA)		The pointer to the neutronics variable array containing the matrix of possible time steps and their powers used for the amplitude equation solution. Shares common with A(LF).
LTABPH	INTCOM (IA)		The pointer to the neutronics variable array containing the secondary pointers to the high table values for background shielding factors for each isotope.
LTABPL	INTCOM (IA)		The pointer to the neutronics variable array containing the secondary pointers to the low table values for background shielding factors for each isotope.
LTABTN	INTCOM (IA)		The pointer to the neutronics variable array containing the secondary pointers to the high table values for temperature shielding factors for each isotope.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LTABTL	INTCOM (IA)		The pointer to the neutronics variable array containing the secondary pointers to the low table values for temperature shielding factors for each isotope.
LTFX	FWBGN1		Length of the cell-wise flux array for a group.
LTOXS	FWBGN1		Length of the cell-wise cross-section array for a group.
LTPC	FWBGN1		Length of the cell-wise precursor concentration array for a delayed-neutron group.
LTPK	FWBGN1		Length of the amplitude equation parameter block.
LTYFC	INTCOM (IA)		The pointer to the neutronics variable array containing the thermal yield per fission and per neutron capture.
LVEL	INTCOM (IA)		The pointer to the neutronics variable array containing the neutron energy group mean velocities.
LVREAT	INTCOM (IA)		The pointer to the neutronics variable array containing the current and previous values of reactivity divided by generation time.
LW	INTCOM (IA)		The pointer to the neutronics variable array containing direction weights for the S_N quadrature.
LWE	INTCOM (IA)		The pointer to the neutronics variable array containing the product of A(LW) and A(LCE).
LWM	INTCOM (IA)		The pointer to the neutronics variable array containing the product of A(LW) and A(LCM).
LXMEM	FWBGN1		Number of words in the LCM expansion for the variable memory option.
LXR	INTCOM (IA)		The pointer to the neutronics variable array containing input radial neutronics-mesh boundaries.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
LXSBUF	INTCOM (IA)		The pointer to the neutronics variable array used as a buffer for input of isotopic cross-sections.
LXSRQT	FWBGN1		File length of the cross-section random file, ISOTXS.
LYH	INTCOM (IA)		The pointer to the neutronics variable array containing the input axial neutronics mesh spacing.
LYR	INTCOM (IA)		The pointer to the neutronics variable array containing the input axial neutronics mesh boundaries.
MATID	S2DIR		Array containing the material identification numbers for the SESAME EOS.
MATXRC	INTCOM (KOM)		Matrix line inversion flag. Row/column = 0/1.
MAXECS	FWBGN2		Maximum length of large core available to the problem.
MAXLEN	FWBGN2		Maximum length of the main neutronics data block A.
MAXPSO	INTCOM (IA)		The maximum number of background shielding factors per isotope accepted by the shielding factor input routine, INFF.
MAXTPT	INTCOM (IA)		The maximum number of temperature shielding factors per isotope accepted by the shielding factor input routine, INFF.
MDAT	KLC1		Tabular EOS array.
MITCRC	INTCOM (KOM)		Input variable to optimize the number of MATSOL iterations. Defined on Fluid-Dynamics Input Card No. 6.
MITMAX	INTCOM (KOM)		Maximum permitted number of MATSOL iterations.
MLINE	PSEDIT		Optional index for functioning of the LINE array.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
MM	INTCOM (IA)		$[ISN * (ISN + 2)]/8$ - number of directions per octant; used in the transport theory option.
MM	IXCHA		Dummy index used in the conversion of a liquid-energy component to a material component; used by the fluid-dynamics modules.
MNR36	PSEDIT		Maximum number of records on the fluid dynamics postprocessor file TAPE36.
MONO	INTCOM (KOM)		Polyatomic gas flag for evaluating the vapor thermal conductivity. See Eqs. (III-80) and (III-81).
MRHS	INTCOM (KOM)		Large-memory pointer after LECCELL and LOCRHS are added together.
MS	INTCOM (IA)		The number of input isotopic mixture instructions.
MT	INTCOM (IA)		The total number of input materials defined.
MTEM	IXCHA		Structure surface index. $MTEM = 0/1/2$ = no structure/first surface/second surface.
MUSTPR	INTCOM (KOM)		Number of cells not converging in the pressure iteration (obsolete).
MUSTR1	INTCOM (KOM)		Number of cells not converging in the vapor density iteration (obsolete).
MUSTR2	INTCOM (KOM)		Number of cells not converging in the liquid density iteration (obsolete).
MXNPAR	FWBGN2		Maximum number of parameter regions. Currently set at 150.
MXTEOS	FWBGN1		Number of LCM words needed for the tabular EOS.
NA	INTCOM (IA)		Integer equivalent of the A array.
NAME	FWBGN1		Problem title.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
NC	COMINP		Number of material components.
NCAN	IXCHA		Flag for the presence of can wall. Can wall/no can wall = 1/0.
NCAT	IXCHA		Structure energy component on can-wall surface NCAT = 0/2/4/5 = no surface/refrozen fuel/can wall/control.
NCAXI	COMINP		The pointer to the neutronics variable array containing the number of subdivisions used to transform the axial fluid-dynamics mesh into the neutronics mesh.
NCELPT	INTCOM (IA)		The number of neutronics mesh cells for which a detailed edit of the shielding factor calculational results is desired.
NCFL	INTCOM (KOM)		The number of components in the FL array.
NCFS	INTCOM (KOM)		The number of components in the FS array.
NCGC	INTCOM (KOM)		The number of materials that can potentially vaporize or condense in a mesh cell. (NCGC = 4).
NCGR	INTCOM (KOM)		The number of vapor-field density components.
NCHT	IXCHA		Structure energy component for a pin surface. NCHT = 0/1/3/5 = no surface/fabricated fuel/cladding/control.
NCL	IXCHA		Flag for the presence of cladding. Cladding/no cladding = 1/0.
NCLC	IXCHA		Maximum number of liquid energy components participating in the multicomponent vaporization/condensation model.
NCLE	INTCOM (KOM)		The number of liquid-field energy components.
NCLEM2	INTCOM (KOM)		The maximum number of components that can vaporize or condense, NCLE - 2.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
NCLR	INTCOM (KOM)		The number of liquid-field density components.
NCON	IXCHA		Flag for the presence of structure control material. Control material/no control material = 1/0.
NCONCL	IXCHA		If NCONCL = 1, crust on cladding is so thin it is only a thermal resistance in the vaporization/condensation model.
NCONCN	IXCHA		If NCONCN = 1, crust on can wall is so thin it is only a thermal resistance in the vaporization/condensation model.
NCQL	INTCOM (KOM)		The number of liquid-field energy components participating in heat transfer (currently NCQL = NCLE).
NCQS	INTCOM (KOM)		The number of structure-field energy components participating in heat transfer.
NCRAD	COMINP		The pointer to the neutronics variable array containing the number of subdivisions used to transform the radial fluid-dynamics mesh into the neutronics mesh.
NCSC	IXCHA		Total number of structure surfaces present in a cell.
NCSE	INTCOM (KOM)		The number of structure-field energy components.
NCSR	INTCOM (KOM)		The number of structure-field density components.
NCSSPR	FWBGN1		Number of sectors required for a single group of a single macroscopic cross section for the full spatial mesh.
NCVC	IXCHA		The actual number of liquid-field energy components that vaporize or condense in a mesh cell.
NCYC	TIME		Number of fluid dynamics time steps.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
NCYCPR	PSEDIT		Cycle number for obtaining a fluid dynamics long print (specified on restart only).
NDELAY	KINET		Number of delayed neutron groups.
NDEST	IXCHA		Integer variable used in logic (with NCONCL) to determine what pin surface material participates in the vaporization/condensation algorithm.
NDKGRP	KINET		Number of decay heating groups.
NDTO	INTCOM (KOM)		Number of cycles for which the time step is maintained at its initial value.
NDUMPI	UNITS		Logical unit 8, file for reading restart data.
NDUMPO	UNITS		Logical unit 21, initial file for writing restart data.
NENLC1	IXCHA		Array giving the initial liquid density component in each liquid energy group.
NENLC2	IXCHA		Array giving the final liquid density component in each liquid energy group.
NENSC1	IXCHA		Array giving the initial structure density component in each structure energy group.
NENSC2	IXCHA		Array giving the final structure density component in each structure energy group.
NERROR	LOCAL		Parameter input error indicator.
NEUF	FWBGN1		File name for overlays 2,0; 2,1; 2,3; 2,5; and 2,6.
NEULPR	PSEDIT		Neutronics long edit output flag.
NEUSPR	SPEDIT		Neutronics short edit output flag.
NEUSTP	INTCOM (KOM)		Flag to indicate if a previous fluid-dynamics time step completed a neutronics time step.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
NEXTER	LOCAL		Indicator to begin next problem.
NEXTRA	UNIT		Logical unit 63, file to simulate ENCODE/ DECODE.
NF1	IXCHA		Flag for fabricated fuel. Fabricated fuel exists/does not exist = 1/0.
NF2	IXCHA		Flag for refrozen fuel. Refrozen fuel exists/ does not exist = 1/0.
NFILM	UNIT		Logical unit 12, file for film output.
NFPLT	UNIT		Logical unit 6, file for integral data output to be post-processed.
NHT	IXCHA		Integer array used to define which struc- ture/structure heat-transfer paths exist.
NINP	UNIT		Logical unit 10, file for problem code dependent input.
NITCRC	INTCOM (KOM)		The minimum number of time steps between reevaluating which direction of line in- version gives the fastest matrix solution convergence rate.
NITMAX	INTCOM (KOM)		Maximum number of allowed pressure itera- tions per cycle.
NITNO	IXCHA		Outer iteration counter. Main use is in the simple vaporization/condensation model.
NITNO2	IXCHA		NITNO2 = MOD (NITNO,2). Used in connec- tion with the acceleration scheme de- scribed by Eq. (IV-116).
NITOVN	INTCOM (IA)		The number of iterations of the rebalance matrix solution (for the transport theory option) or the flux matrix solution (for the diffusion theory option) between up- dates of the over-relaxation factor.
NITSR2	SOLVE		Two times NITSRU. Used in logic for the AIM algorithm in subroutine MATSOL.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
NITSRU	INTCOM (KOM)		Input parameter controlling convergence in subroutine MATSOL. See Fluid-Dynamics input Card No. 6.
NK	KINET		NK = 9, number of point kinetics variables per mesh cell.
NKI	KINET		Number of point kinetics variables for one full row of mesh cells.
NLC	FWBGN1		Length of large core required for the fluid dynamics mesh cell variables.
NLHT	IXCHA		NCLE - 1. Used in the liquid-liquid heat transfer logic.
NMAT	INTCOM (KOM)		NMAT = 5, the number of different materials.
NMATM1	INTCOM (KOM)		NMAT - 1. Used for summations over materials when the fission gas component is to be skipped.
NN	INTCOM (IA)		ISN/2.
NODE	PSEDIT		The number of cell locations to be edited in a long print.
NODES	INTCOM (KOM)		Specifies the cell locations to be printed during a long print.
NOFS	UNITS		Dummy file number for paper and/or film output.
NORDO	INTCOM (IA)	I_0	The maximum number of initial unaccelerated outer (fission-source) iterations.
NORDM	INTCOM (IA)		Largest order of Chebyshev accelerating polynomial permitted.
NOSET	NITIAL		The number of mesh cell sets to be specified for the problem. Each mesh cell set defines the composition in a rectangular area of mesh cells.
NOUT	UNITS		Logical unit 9, file for problem output.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
NP1	IXCHA		$N + 1$. Used to economize in DO LOOP logic.
NPAR	FWBGN2		The maximum number (currently 28) of parameters in each parameter region.
NPAREG	INTCOM (KOM)		The number of input geometry parameter regions; NPAREG $\leq$ 150.
NPLT3D	UNITS		Logical unit 36, the file for fluid dynamic state variables.
NPRLPF	PSEDIT		A neutronics logical print flag.
NPT1	IXCHA		Iteration counter on the Newton-Raphson iteration for droplet vapor-side heat-transfer coefficients.
NPT2	IXCHA		Iteration counter on the Newton-Raphson iteration for structure vapor-side heat-transfer coefficients.
NQ	INTCOM (KOM)		Number of fluid dynamics variables per mesh cell.
NQ2	INTCOM (KOM)		$NQ + NQ$.
NQ2L	INTCOM (KOM)		$NQL + NQL$.
NQI	INTCOM (KOM)		$NQ \cdot IP^2$, the number of words for one full row of mesh cells.
NQI2	INTCOM (KOM)		$NQI + NQI$.
NQL	INTCOM (KOM)		$NQ \cdot (1 - LCM)$.
NR36	PSEDIT		Number of records written to the fluid dynamics postprocessing file.
NRHS	INTCOM (KOM)		Array number for variable that is being read into/out of LCM during the density component update.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
NRS	S2DIR		Number of SESAME EOS regions.
NRXS	INTCOM		Number of cross-section regions.
NS	IXCHA		Flag for finding a clever surface index in the multicomponent phase-transition model.
NSECT	INTCOM (IA)		LCM pointer of XS data.
NSHT	IXCHA		NCSE - 1. Used in DO LOOP initialization when considering pairs of structural materials.
NSS	IXCHA		Similar to NS.
NSTAT	STAT		Number of words in the STAT common block.
NTMPIN	INTCOM (KOM)		The number of entries in the table for sodium inlet temperature vs time.
NTR	IXCHA		Local variable for type of cell geometric configuration.
NUMINC	INTCOM (KOM)		The number of incompressible mesh cells.
NUMIT	INTCOM (KOM)		Number of pressure iterations during the current cycle.
NUMITP	INTCOM (KOM)		Number of pressure iterations on the previous cycle.
NUMRO	INTCOM (KOM)		Number of density component iterations (currently obsolete).
NUMTD	TIME		Number of restart dumps.
NXCYC	PSEDIT		The number of time steps between long prints before time DTOC. If NXCYC = 0, long prints are done every DTO.
NXTCRC	INTCOM (KOM)		Cycle counter for the NITCRC control parameter.
NYCYC	PSEDIT		Cycle counter for the NXCYC print control.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
NYLD	INTCOM (KOM)		Number of tabular data points relating the yield stress to temperature.
OITNO	INTCOM (IA)		The current outer iteration number.
OHTC	RXCHA	$(\frac{r_{pm}}{k_{Lm}} + \frac{r_{pk}}{k_{Lk}})^{-1}$	Basic component for the liquid-liquid heat-transfer relationship [Eq. (III-132)].
OHTCL	RXCHA	$15\alpha_{Lk} \frac{k_{Lk}}{r_{pk}^2}$	Temperature coefficient in liquid-liquid heat-transfer limit [Eq. (III-133)].
OM	INTCOM (IA)		Inverse period.
OMEG	SIMCOM (IA)		The overrelaxation factors for the current energy group.
OMR	SIMCOM (IA)		Inverse period for the previous reactivity time step.
ORCO	SIMCOM (KOM)	C_{ORF}	The orifice coefficient for the pressure drop at location given by node LCCORF.
ORV	SIMCOM (IA)		The inverse period divided by the mean velocity of the current energy group.
OTEMP	RXCHA		Array of operational temperature(s) used in the multicomponent phase-transition model.
OUTF	FWBGN2		File name for overlay 2,2.
P	SIMCOM (KOM)	p	The mesh-cell pressure.
PALPHG	SIMCOM (KOM)	$\tilde{p}^{\kappa=0} \alpha_G$	Product of the vapor volume fraction and the initial end-of-cycle pressure estimate.
PAREG	SIMCOM (KOM)		A parameter region flag.
PARPL	EOS	$p_{vM(m)}(T_{Lm})$	The partial pressure contribution coming from the vapor pressure of a given liquid-field component.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
PARTL	TIME		Fractional part of task time limit to use for problem execution, remainder of the time is used for cleanup.
PARVIS	SIMCOM (KOM)	μ_p	The particle viscosity coefficient multiplier.
PBOT	SIMCOM (KOM)		Pressure entries in the pressure-vs-time table for the bottom boundary.
PBOTT	SIMCOM (KOM)		Pressure at the bottom boundary at time t.
PCRIT	SIMCOM (KOM)	$P_{Crt,M}$	The critical pressure for a given EOS material.
PCURIN	SIMCOM (IA)	I^{S-1}	The leakage contribution to the reactivity in the previous flux-shape time step. Used for the transport theory option only.
PDELV	RXCHA		Pressure volume work done on the vapor field when intergranular fission gas is released.
PDIF	RXCHA	$p_{vm}(T_{Lm}) - p$	The difference between the component vapor pressure and the ambient pressure for use in the flashing criterion for determining droplet radii.
PDIFF1	RXCHA	$ p_{ij} - p_{i+1,j} $	Radial absolute pressure change in the positive direction for use in Eq. (IV-53).
PDIFF2	RXCHA	$ p_{ij} - p_{i-1,j} $	Radial absolute pressure change in the negative direction for use in Eq. (IV-53).
PHF	FWBGN2		File name for overlay 3,1.
PHI	SIMCOM (IA)		The normalized power amplitude from the space-dependent neutronics calculation.
PHI	KINET		The normalized power amplitude from the point kinetics calculation.
PHIG2	RXCHA	ϕ_G^2	Two-phase friction-factor multiplier for vapor.
PHIINF	RXCHA	ϕ_{mk}	Component of the mixture rule for viscosity and thermal conductivity given by Eq. (III-87).

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
PHIL2	RXCHA	Φ_L^2	Two-phase friction-factor multiplier for liquid.
PHIP	SIMCOM (IA)		The initial value of the amplitude; used as an initial condition for the current solution interval of the amplitude equation. Used for the space-dependent neutronics calculation.
PHIP	KINET		The initial value of the amplitude; used as an initial condition for the current solution interval of the amplitude equation. Used for the point kinetics solution.
PHIPS	SIMCOM (IA)		The amplitude at the beginning of the current flux-shape time step.
PIEE	SIMCOM (KOM)	λ	Particle interference effect exponent. See Eq. (III-166).
PINT	SIMCOM (IA)		The current integral of the power. Used for the space-dependent neutronics calculation.
PINTG	SIMCOM (IA)		The current integral of the amplitude function. Used for the space-dependent neutronics calculation.
PINTG	KINET		The current integral of the power. Used for the point kinetics calculation.
PINTGO	SIMCOM (IA)		The difference between the energy deposition in the space-dependent neutronics solution and the energy deposition in the fluid-dynamics solution.
PINTGO	KINET		The difference between the energy deposition in the space-dependent neutronics solution and the energy deposition in the fluid-dynamics solution.
PINTPS	SIMCOM (IA)		Normalized amplitude-function integral at the beginning of the current flux shape time step.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
PINTS	SIMCOM (IA)		The value of the power integral at the beginning of the current flux-shape.
PKDF	FWBGN2		File name for overlay 2,4.
PLUGM	SIMCOM (KOM)		Mass of movable plug (input).
PNI	NITIAL	p^i	Initial vapor-field pressure.
PNOFL	SIMCOM (KOM)	P_{nf}	Pressure in the region IREG volume fraction not available for flow.
POW	KINET	P_o	The total initial power of the system. Used for the point kinetics solution only.
POWDK	SIMCOM (IA)		Decay heat power amplitude. Not used in the current version of SIMMER-II.
POWDKP	SIMCOM (IA)		Previous decay heat power amplitude. Not used in the current version of SIMMER-II.
POWDKP	KINET		Decay heat power amplitude on the previous time step. Not used in the current version of SIMMER-II.
POWER	SIMCOM (IA)	P_o	Initial power.
POWFS	SIMCOM (IA)	N^r	The normalized amplitude function used for the space-dependent neutronics solution.
POWFS	KINET		The normalized power. Used for the point-kinetics neutronics solution.
POWFSP	SIMCOM (IA)	N^{r-1}	The initial value of the amplitude function over the current solution interval of the amplitude equation. Used for the space-dependent neutronics solution.
POWFSP	KINET		The initial value at the normalized power over the current solution interval of the point-kinetics equations. Used for the URANUS option.
POWPS	SIMCOM (IA)	N^{S-1}	The amplitude function at the beginning of the current flux-shape time step.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
PRBL	RXCHA		Prandtl number in the boundary layer for a surface in the multicomponent phase-transition model.
PRBNDY	SIMCOM (KOM)		Variable containing the right-hand side boundary pressure, for the pressure-time option.
PREATB	KINET		Tabulated values of driving reactivity or power. Used for the URANUS option.
PREATM	KINET		Tabulated times for PREATB. Used for the URANUS option.
PRG	RXCHA	Pr_G	Vapor-field Prandtl number for a SIMMER-II mesh cell.
PRIGHT	SIMCOM (KOM)		Pressure entries in the pressure-time table for the right boundary.
PRINLF	PSEDIT		A logical variable indicating the choice of neutronics print options for the film file in the space-dependent neutronics output.
PRINLP	PSEDIT		A logical variable indicating the choice of neutronics print options for the film file in the space-dependent neutronics output.
PRINSF	PSEDIT		A logical variable indicating the choice of neutronics print options for the film file in the space-dependent neutronics output.
PRINSP	PSEDIT		A logical variable indicating the choice of neutronics print options for the print file in the space-dependent neutronics output.
PRL	RXCHA	Pr_{Lm}	Prandtl number for liquid-field energy component m.
PSELAS	PSEDIT		Last entry of common block PSEDIT.
PSHAPR	PASS		Unnormalized radial shape of the internal energy source.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
PSHAPZ	PASS		Unnormalized axial shape of the internal energy source.
PSTAR	SIMCOM (KOM)	p_m^*	Parameter in the vapor pressure expression for vapor-field material component m.
PTOP	SIMCOM (KOM)		Pressure entries in the pressure-time table for the top boundary.
PTOPT	SIMCOM (KOM)		The pressure at the top boundary of the problem at time t.
PTPAVE	SIMCOM (KOM)		Peak average pressure on the head for use in a post-disassembly expansion problem.
PTPMAX	SIMCOM (KOM)		Actual peak pressure on the head at location IPTMAX.
PV	EOS		PV(1) is the pressure returned from P4. PV(2) is the pressure derivative with respect to vapor density.
PVDENT	PASS		Temporary array used in the pressure iteration to avoid recalculation of terms for the iteration parameter.
Q	SOLVE		Array containing the right-hand sides of the equation set to be solved by MATSOL.
QG	SIMCOM (KOM)	Q_G	The total energy transfer to the vapor per unit volume as computed in subroutine XCHANJ.
QGC	RXCHA	$q_{IGmLm}^{\kappa+1}$	Heat flow to the vapor field from the interface with liquid-field energy component m.
QGS	RXCHA	$q_{IGmSk}^{\kappa+1}$	Heat flow to the vapor field for material component m resulting from the interface with structure energy component k.
QGWORK	SIMCOM (KOM)	Q_G	Total energy transfer to the vapor per unit volume.
QL	SIMCOM (KOM)	Q_{Lm}	The total energy transfer to the liquid-field energy component because of heat and mass transfer.

TABLE J-I (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
QLC	RXCHA	$q_{ILmGm}^{\kappa+1}$	Heat flow to liquid energy component m from the interface with the vapor field.
QMULT	COMINP		The normalized power integral over the current reactivity step from the fluid-dynamics solution.
QMULTP	SIMCOM (IA)		The difference between the neutronic and fluid dynamics values for the normalized power integral during the reactivity update.
QMULTS	SIMCOM (IA)		The normalized power integral from the fluid-dynamics solution.
QS	SIMCOM (KOM)	Q_{Sm}	The total energy transfer to the structure-field energy component m because of heat and mass transfer.
QSC	RXCHA	q_{ISkGm}^{κ}	Heat flow to structure energy component k from the interface with vapor material component m.
QSCANW	RXCHA		Heat flow to structure material NCAT before the vaporization/condensation operations are performed.
QSCLAD	RXCHA		Heat flow to structure material NCHT before the vaporization/condensation operations are performed.
QT	SIMCOM (IA)		A temporary variable used to transfer the source term to the FIXUP routine in the transport theory option.
QT	SOLVE		Temporary storage array for various quantities used in the pressure iteration.
QTRAN	RXCHA		Temporary array containing the energy transfer rates associated with mass transfer during melting or freezing determinations.
R1MTHS	RXCHA	$(1 - \alpha_S)^{-1}$	Recalculation of the reciprocal of the fluid volume fraction in subroutine PHASE.
RADT	SOLVE		Reciprocal of ADT.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RAMPTO	SIMCOM (IA)		Time at which a programmed reactivity ramp is to start.
RAREA	SIMCOM (KOM)		Radial area for a mesh cell, given by $(2\pi r_i)^{\frac{1}{2}} \Delta r_i$.
RAREAI	SIMCOM (KOM)		Radial area for a momentum cell, given by $(2\pi r_{i+1/2})^{\frac{1}{2}} \Delta r_{i+1/2}$.
RATMSQ	SIMCOM (KOM)	$(\frac{1}{\sigma_m})^2$	Inverse of the molecular diameter squared. Used in computation of vapor mixture properties.
RCFLR	RXCHA		Value of RFLR or CFLR actually used in material melting.
RCLAD	SIMCOM (KOM)	r_{clad}^i	The input outer radius of the fuel pin cladding.
RCONT	SIMCOM (KOM)		The flag for continuative outflow on the right boundary.
RCOURT	SIMCOM (KOM)	C_t^{-1}	Reciprocal of COURNT.
RCRIT	SIMCOM (KOM)	$R_{Crt,M}$	Gas "constant" at the critical point. Defined following Eq. (III-55).
RCVL	SIMCOM (KOM)	$1/c_{vLM}$	The reciprocal of the heat capacity for liquid-field material component M.
RCVS	SIMCOM (KOM)	$1/c_{vSM}$	The reciprocal of the heat capacity for structure-field material component M.
RDH	RXCHA	$1/D_h$	Reciprocal of the hydraulic diameter.
RDPDRG	SIMCOM (KOM)	G^{-1}	The derivative of a macroscopic density with respect to pressure for constant internal energy.
RDR	SIMCOM (KOM)	$\frac{1}{\Delta r_i}$	The inverse of the radial mesh spacing.
RDRIP	SIMCOM (KOM)	$\frac{1}{\Delta r_{i+1/2}}$	The inverse of the radial mesh spacing of a momentum cell.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RDT	SIMCOM (KOM)	$\frac{1}{\Delta t}$	The inverse of the time step increment.
RDTN	TIME	$\frac{1}{\Delta \tilde{t}}$	The inverse of the estimated fluid dynamics time step.
RDTT	RXCHA	$\frac{1}{\Delta \tilde{t}}$	Temporary variable that equals RDTN in the calculation of the exchange functions.
RDZ	SIMCOM (KOM)	$\frac{1}{\Delta z_j}$	The inverse of the axial mesh spacing.
RDZIP	SIMCOM (KOM)	$\frac{1}{\Delta z_{j+1/2}}$	The inverse of the axial mesh spacing of a momentum cell.
REACIN	KINET		The input initial reactivity for a point kinetics calculation.
REACOL	KINET	$\delta \rho_{NA}$	Sodium displacement reactivity feedback for a point kinetics calculation.
REACTRT	SIMCOM (IA)		Reactivity insertion rate.
REACT	SIMCOM (IA)		Current value of the total reactivity.
REACTI	SIMCOM (IA)		The difference between the input initial reactivity and the results of the initial reactivity calculation.
REACTL	KINET	$\delta \rho_{Cont}$	Control material displacement reactivity feedback for a point kinetics calculation.
REACTS	SIMCOM (IA)		Reactivity at the beginning of the flux-shape time step.
READOP	KINET	$\delta \rho_{Dop}$	Doppler reactivity feedback for a point kinetics calculation.
REAFER	KINET	$\delta \rho_{Fert}$	Fertile fuel displacement reactivity feedback for a point kinetics calculation.
REAFIS	KINET	δr_{Fiss}	Fissile fuel displacement reactivity feedback for a point kinetics calculation.
REAPGM	KINET	$\delta \rho_p$	Driving reactivity for a point kinetics calculation.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
REASTR	KINET	$\delta\rho_{SS}$	Stainless steel displacement reactivity feedback for a point kinetics calculation.
REATIM	TIME		The time of the current reactivity step.
REATOT	KINET	ρ	Total reactivity.
REG	RXCHA	Re_G	Reynolds number for the vapor field in a cell.
REL	RXCHA	Re_{Lm}	Reynolds number for liquid-field energy component m.
RELG	RXCHA	Re_{GLm}	Reynolds number used for vapor flow around a sphere of liquid-field energy component m.
RELT	RXCHA	Re_L	Reynolds number used in liquid/structure momentum transfer. Defined by Eq. (III-143).
RERHOL	SIMCOM (KOM)		Temporary variable used to evaluate Eq. (IV-25) for the input mass conservation algorithm.
RFLMX	SIMCOM (IA)		The variable used to scale the flux so that the L_2 norm calculation used for the matrix solution does not exceed the computer range limitation.
RFLR	SIMCOM (KOM)		The maximum mass-transfer rate of failed structure to the liquid or vapor field.
RGBR	SIMCOM (KOM)	$\bar{\rho}_{gm}$	The macroscopic density of the vapor-field density components.
RGBRI	NITIAL	$\bar{\rho}_{gm}^0$	The initial macroscopic density of the vapor-field density components.
RGBRT	SIMCOM (KOM)	$\bar{\rho}_{gm}$	The intermediate value of the end-of-time step macroscopic density for vapor-field density components.
RGGAM	RXCHA	$\sum_m \bar{\rho}_{Gm} R_{M(m)}$	Temporary variable used in the DTL(11) time step control, Eq. (IV-260).

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RGRAIN	SIMCOM (KOM)	r_{gr}	The effective radius of the fuel grains in unstructured fuel.
RGTYP	SIMCOM (KOM)		Geometric configuration indicator (see input description).
RHFUS	EOS	$(h_{f,M})^{-1}$	Inverse of the heat of fusion for material M.
RHOA1	SIMCOM (KOM)	a_{1m}	First coefficient for the liquid density vs temperature equation at low temperature.
RHOA2	SIMCOM (KOM)	a_{2m}	Second coefficient for the liquid-density-temperature equation at low temperature.
RHOA3	SIMCOM (KOM)	a_{3m}	Third coefficient for the liquid-density-temperature relationship at low temperature.
RHOA4	SIMCOM (KOM)	a_{4m}	Fourth coefficient for the liquid-density-temperature relationship at low temperature.
RHOB1	SIMCOM (KOM)	b_{1m}	First coefficient for the liquid-density-temperature relationship at high temperature.
RHOB2	SIMCOM (KOM)	b_{2m}	Exponent for the liquid-density-temperature relationship at high temperature.
RHOB3	SIMCOM (KOM)	b_{3m}	Last coefficient for the liquid-density-temperature equation at high temperature.
RHODOL	TIME		The most recent value of A(LVREAT) divided by the total effective delayed neutron function.
RHODTL	TIME		Maximum fluid-dynamics time step if reactivity exceeds RHOLIM. Affects DTL(9).
RHOIN	SIMCOM (KOM)	ρ_t^i	Input initial reactivity for a space-dependent neutronics calculation.
RHOLIM	TIME		Reactivity above which RHODTL limits the fluid-dynamics time step.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RHOLMI	SIMCOM (KOM)	ρ_{Lm}	Temperature-dependent microscopic density of liquid-energy component m.
RHOSAT	SIMCOM (KOM)		Saturated grain boundary fission gas concentration. Input in atoms per square meter of grain boundary surface. Stored as kilograms of gas per cubic meter of un-restructured fuel.
RHS	SOLVE		Array to transfer component macroscopic densities to/from LCM. These are the right-hand sides of Eq. (IV-219).
RHTRAN	RXCHA	$H_{eff,mk}$	A variable used to contain the indicated effective heat-transfer coefficient in the solution of Eq. (IV-70).
RHVO	RXCHA	$(H_{GLm}^*)^{-1}$	The inverse of HVO.
RHVS0	RXCHA	$(h_{Gsk}^*)^{-1}$	The inverse of HVSO.
RI	SIMCOM (KOM)	r_i	Radius of the center of mesh cell i.
RIBAR	SIMCOM (KOM)	$1/\bar{I}$	Reciprocal of FIBAR.
RIBJB	SIMCOM (KOM)	$1/(\bar{I} \cdot \bar{J})$	Used in control region execution.
RINPC	COMINP		Array to which all variables in common block COMINP are equivalenced.
RIP	SIMCOM (KOM)	$r_{i+1/2}$	Radius of the right side of mesh cell i.
RJBAR	SIMCOM (KOM)	$(1/\bar{J})$	Reciprocal of FJBAR.
RL	EOS	ρ_{Lm}	Microscopic density of liquid-field energy component as calculated by the EOS. Not used in SIMMER-II.
RLBR	SIMCOM (KOM)	$\bar{\rho}_{Lm}$	The macroscopic density of the liquid-field density components.
RLBRI	NITIAL	$\bar{\rho}_{Lm}^0$	The initial macroscopic density of the liquid-field density components.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RLBRT	SIMCOM (KOM)	$\bar{\rho}_{lm}$	The intermediate value of the end-of-time time-step macroscopic density for liquid-field density components.
RLBRTL	RXCHA	$\bar{\rho}_{L1}$	A temporary variable containing the total macroscopic density of the first liquid energy component (liquid fuel).
RLFLM	TIME		The cumulative liquid-field component density change since the previous postprocessor dump.
RLL	SIMCOM (KOM)	R_{mk}	The user-specified weights on the heat transfer coefficients.
RLMOV	TIME		The maximum liquid-field component density change between postprocessor dumps.
RMAT	SIMCOM (KOM)	R_M	The "gas constant" for material M. Evaluated from Eq. (III-54).
RMUL	RXCHA		Temporary variable containing the effective viscosity for evaluating RELT. This is the denominator of Eq. (III-143).
ROBR1	SIMCOM (KOM)	$\bar{\rho}_{Fert}$	Current value of the fertile fuel total macroscopic density.
ROBR2	SIMCOM (KOM)	$\bar{\rho}_{Fiss}$	Current value of the fissile fuel total macroscopic density.
ROBR3	SIMCOM (KOM)	$\bar{\rho}_{SS}$	Current value of the stainless steel total macroscopic density.
ROBR4	SIMCOM (KOM)	$\bar{\rho}_{Na}$	Current value of the sodium total macroscopic density.
ROBR5	SIMCOM (KOM)	$\bar{\rho}_{Cont}$	Current value of the control material total macroscopic density.
ROBR10	SIMCOM (KOM)	ρ_{Fert}	Initial value of the fertile fuel total macroscopic density for a point kinetics calculation.
ROBR20	SIMCOM (KOM)	$\bar{\rho}_{Fiss}^0$	Initial value of the fissile fuel total macroscopic density for a point kinetics calculation.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ROBR30	SIMCOM (KOM)	$\bar{\rho}_{SS}^0$	Initial value of the stainless-steel total macroscopic density for a point kinetics calculation.
ROBR40	SIMCOM (KOM)	$\bar{\rho}_{NA}$	Initial value of the sodium total macroscopic density for a point kinetics calculation.
ROBR50	SIMCOM (KOM)	$\bar{\rho}_{Cont}$	Initial value of the control material total macroscopic density for a point kinetics calculation.
ROCELL	RXCHA		Free steam vapor density in the multi-component, vaporization/condensation model.
ROG	EOS	$\bar{\rho}_G/\alpha_G$	The vapor-field effective microscopic density.
ROGBR	SIMCOM (KOM)	$\bar{\rho}_G$	The sum of the vapor-field component macroscopic densities.
ROGBRI	NITIAL	$\bar{\rho}_G^i$	The initial sum of the vapor-field component macroscopic densities.
ROGBRT	SIMCOM (KOM)	$\bar{\bar{\rho}}_G$	The intermediate value of the end-of-time-step vapor-field total macroscopic density.
ROGCRT	SIMCOM (KOM)	$\rho_{Crt,M}$	The critical density (microscopic) of material M.
ROGCUT	SIMCOM (KOM)		The minimum vapor-field component macroscopic density, $\bar{\rho}_{gm}$, used for time-step controls and phase-transition tests.
ROGMXV	SIMCOM (KOM)		Dummy input variable on fluid-dynamics Card No. 67. Currently overwritten by 10 ²⁰ .
ROGI	NITIAL	ρ_G^i	The initial value of the vapor-field effective microscopic density.
ROGP95	SIMCOM (KOM)		The saturated vapor density of material M when $TSAT = 0.95 * TCRIT$.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ROGP	RXCHA	$\bar{\rho}_{Gm}^n$	Temporary value of the beginning of time step vapor macroscopic density used in the vaporization/ condensation algorithm.
ROGPM	EOS	ρ_{Gm}	Microscopic density for vapor component m.
ROGPP	RXCHA	$\bar{\rho}_{Gm}^k$	Value of the vapor macroscopic density after k iterations in the simple vaporization/ condensation model.
ROGPPP	RXCHA		Value of ROGPP on the previous iteration.
ROGPRU	RXCHA	$\bar{\rho}_G^{k+1}$	The converged vapor macroscopic density from the vaporization/condensation model.
ROL	SIMCOM (KOM)	ρ_{Lm}	The microscopic density of the liquid-field density components.
ROLBR	SIMCOM (KOM)	$\bar{\rho}_L$	The sum of the liquid-field component macroscopic densities.
ROLBRI	NITIAL	$\bar{\rho}_L^i$	The initial sum of the macroscopic densities of the liquid-field components.
ROLBRT	SIMCOM (KOM)	$\bar{\rho}_L$	The intermediate value of the end-of-time-step liquid-field total macroscopic density.
ROLC	RXCHA	ρ_{Lm}	A temporary variable containing microscopic densities of liquid-field energy components.
ROLCRT	SIMCOM (KOM)	$\rho_{Crt,M}$	The critical density obtained by following the liquid-density-temperature curve to the critical temperature.
ROLE	SIMCOM (KOM)		The liquid microscopic density of each material. Currently used only if an actual liquid component is below the solidus energy.
ROLPP	RXCHA	$\bar{\rho}_{Lm}/\Delta t$	Macroscopic liquid density divided by the time step size. A convenience variable used for exchange-function calculations.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ROMUK	RXCHA	$1/(\sigma_m^2 \Omega_{k,m})$	Inverse of the denominator for Eqs. (III-80) and (III-82). Used in vapor property evaluations.
ROPS	RXCHA		A beginning-of-time-step value for the macroscopic density of a surface contributing to the multicomponent vaporization/condensation model.
ROPSN	RXCHA		Updated values of the surface densities used in the multicomponent vaporization/condensation model.
ROS	SIMCOM (KOM)	ρ_{sm}	The microscopic density of the structure-field density components.
ROSE	SIMCOM (KOM)	ρ_{Sm}	The solid microscopic density of each material.
ROSFAL	SIMCOM (KOM)		The subassembly can wall density below which failure is assumed to occur.
ROSLT	SIMCOM (KOM)		Fraction of the available fuel or steel mass that can be melted or frozen by non-equilibrium model in a time step.
ROSMX	RXCHA		Temporary variable, using in melting or freezing, indicating the macroscopic density of the available material.
ROSP	RXCHA	$\bar{\rho}_{Sm}$	Temporary variable containing the macroscopic density for a structure energy component.
ROSPB	RXCHA		Macroscopic structure density for a surface in the simple vaporization/condensation model. Temporary convenience variable.
ROSPPP	RXCHA	$\bar{\rho}_{Sm}/\Delta\tilde{t}$	Macroscopic structure density divided by the time-step size. A convenience variable used for exchange-function calculations.
ROUG	SIMCOM (KOM)	$(\rho_G U_G)_{i+1/2,j}$	The vapor-field radial momentum density.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ROUL	SIMCOM (KOM)	$(\rho_L U_L)_{i+1/2,j}$	The liquid-field radial momentum density.
ROVG	SIMCOM (KOM)	$(\rho_G V_G)_{i,j+1/2}$	The vapor-field axial momentum density.
ROVL	SIMCOM (KOM)	$(\rho_L V_L)_{i,j+1/2}$	The liquid-field axial momentum density.
RP	RXCHA	r_{pm}	The droplet radii for the first four liquid energy components.
RP1	RXCHA	$r_{pm,1}$	The droplet radii determined by the Weber number.
RP2	RXCHA	$r_{pm,2}$	The droplet radii determined by the cell hydraulic diameter.
RP3	RXCHA	$r_{pm,3}$	The droplet radii determined by the internal pressure (flashing criterion).
RP4	RXCHA	$r_{pm,4}$	The droplet radii determined by the available component volume.
RP5	RXCHA	$r_{pm,5}$	The droplet radii determined by droplet coalescence.
RPELL	SIMCOM (KOM)	r_{pell}^i	Outer radius of the fuel pellets.
RPFMAX	SIMCOM (KOM)	C_2	Inverse of the maximum particle packing fraction. Used in Eq. (III-183).
RPLC	SIMCOM (KOM)	r_{pL4}	Droplet radius for liquid control material.
RPLF	SIMCOM (KOM)	r_{pL1}	Droplet radius for liquid fuel.
RPLNA	SIMCOM (KOM)	r_{pL3}	Droplet radius for sodium.
RPLS	SIMCOM (KOM)	r_{pL2}	Droplet radius for liquid steel.
RPMAX	SIMCOM (KOM)	$r_{pm,max}$	Maximum droplet radius allowed.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RPMIN	SIMCOM (KOM)	$r_{pm,min}$	Minimum droplet radius allowed.
RQ1	SIMCOM (IA)		The scaled L_2 norm of the cell-wise flux (for the diffusion option) or rebalance factors (for the transport option) for the most recent solution iterate.
RQ2	SIMCOM (IA)		The scaled L_2 norm of the cell-wise flux (for the diffusion option) or rebalance factors (for the transport option) for the previous solution iterate.
RPS	RXCHA	r_{pm}^n	Value of a droplet radius at the beginning of a time step. Used in the RP5 calculation.
RPSF	SIMCOM (KOM)	r_{psf}	The particle radius for solid fuel in the liquid field.
RPSS	SIMCOM (KOM)	r_{pss}	The particle radius for solid steel in the liquid field.
RPSTAR	SIMCOM (KOM)	$\frac{1}{p_M^*}$	The inverse of the parameter in the vapor pressure expression for vapor-field material component M.
RQTACS	UNITS		Request table for IFSACS random file.
RQTBCS	UNITS		Request table for IFSBCS random file.
RRI	SIMCOM (KOM)	$1/r_i$	Inverse of the radius of the center of mesh cell i.
RRIP	SIMCOM (KOM)	$1/r_{i+1/2}$	Inverse of the radius of the right side of mesh cell i.
RROCSQ	SIMCOM (KOM)	$1/\rho_{\ell m} c_{\ell m}^2$	The inverse of the product of the liquid microscopic density and the square of the sonic velocity.
RROG	RXCHA	$\alpha_G/\bar{\rho}_G$	Reciprocal of ROG.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RROGPS	RXCHA		Temporary variable containing component-independent quantities used in computing the DTL(8) time-step control.
RROGPU	RXCHA	$1/\bar{\rho}_G^{\kappa+1}$	Reciprocal of ROGPRU.
RROL	SIMCOM (KOM)	$1/\rho_{lm}$	The inverse of the liquid microscopic density for each liquid-field density component.
RROLE	SIMCOM (KOM)	$1/\rho_{Lm}$	The inverse of the liquid microscopic density for each liquid-field energy component.
RROLM	SIMCOM (KOM)	$1/\rho_{Lm}$	Temporary cell-dependent inverse of the liquid microscopic density for the first three liquid-field energy components.
RROS	SIMCOM (KOM)	$1/\rho_{sm}$	The inverse of the structure microscopic density for each structure-field density component.
RROSE	SIMCOM (KOM)	$1/\rho_{Sm}$	The inverse of the structure microscopic density for each structure-field energy component.
RRPN	RXCHA	$1/r_{pm}$	Reciprocal of the droplet radius, RP.
RRSCLD	IMCOM (KOM)	$1/(\bar{\rho}_{s5})_{fail}$	Reciprocal of RSCLAD.
RRSENT	RXCHA	$h_{\lambda g, M}^{\kappa-1}$	Reciprocal of RSENT, which is actually SENT from the previous iteration.
RSBR	SIMCOM (KOM)	$\bar{\rho}_{sm}$	The macroscopic density of the structure-field density components.
RSBR6L	RXCHA		Can-wall macroscopic density that can be potentially set over into a cell from the left.
RSBR6R	RXCHA		Can-wall macroscopic density that can be potentially set over into a cell from the right.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RSBR7T	RXCHA		Temporary storage for the control density to be used if a mesh cell type is negative.
RSBRI	NITIAL	$\bar{\rho}_{sm}^i$	The initial macroscopic density of the structure-field density components.
RSBRT5	RXCHA		Temporary storage for the cladding density that can participate in melting or freezing.
RSCLAD	SIMCOM (KOM)	$(\bar{\rho}_{s5})_{fail}$	The minimum structure density at which the full no-flow volume fraction is added to the structure-field volume fraction.
RSCV	RXCHA	$\bar{\rho}_{s6} c_{vs2}$	Temporary storage for can-wall heat capacity per unit volume used in the can wall set-over model.
RSENT	RXCHA	$1/h_{lg,M}$	Reciprocal of SENT.
RSFLM	TIME		The cumulative structure-field component density change since the previous postprocessor dump.
RSMOV	TIME		The maximum structure-field component density change between postprocessor dumps.
RSMPHI	RXCHA		Component weighting factor for evaluating vapor viscosity and thermal conductivity in Eqs. (III-85) and (III-86).
RSURF	RXCHA		An intermediate value for the macroscopic density of a surface used in the multi-component vaporization/condensation model.
RTCRT	SIMCOM (KOM)	$1/T_{Crt,M}$	Reciprocal of a material's critical temperature.
RTHICK	RXCHA		Reciprocal of the frozen material thickness in the structure configuration model.
RTIME	TIME		Most recent real time for the flux-shape calculation.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
RTSTAR	SIMCOM (KOM)	$1/T_M^*$	The inverse of the input temperature in the AEOS vapor pressure equation.
RVT	SIMCOM (IA)		The coefficient of the source portion of the backward difference of the flux-shape time derivative from the quasistatic algorithm.
RWTMOL	SIMCOM (KOM)	$1/A_M$	The inverse of the molecular weight of material M.
RXFFLR	SIMCOM (KOM)	$\frac{1}{(x_{Fk} - f_{Fk})}$	A parameter used with melting of fuel and steel particles so that the remaining solid has a melt fraction of f_{Fk} .
RXFLRM	SIMCOM (KOM)	$\frac{1}{x_{Fk}} - 1$	A term to extract particle mass transfer rates from liquid mass transfer rates. See Table XIII.
S	SIMCOM (IA)		Source used in FIXUP.
S2DLAS	S2DIR		Last word of the S2DIR common block.
SCRC	SIMCOM (KOM)		Structure vaporization/condensation rate coefficient.
SEGO	SIMCOM (KOM)		A summation, by components, of vapor-field component energy transport across the boundaries of the system.
SEI	SIMCOM (KOM)		Initial total energy in the system.
SELO	SIMCOM (KOM)		A summation, by components, of liquid-field component energy transport across the boundaries of the system.
SENT	EOS	$h_{lg,M}$	The material enthalpy of vaporization.
SIEFLR	RXCHA	e_{Fk}	Breakup energy for component k given by Eq. (III-191).
SIEG	SIMCOM (KOM)	e_G	The specific internal energy of the vapor for a mesh cell.

TABLE J-1 (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
SIEGD	RXCHA	$ \tilde{e}_G^{n+1} - e_G^n $	Energy difference used in the calculation of DTL(11).
SIEGI	NITIAL	e_G^i	The initial specific internal energy of the vapor for a mesh cell.
SIEGL	EOS		In P4, SIEGL is the input vapor internal energy. In GE4, SIEGL is the output vapor internal energy.
SIEGS	EOS	$e_{\text{Vap},M}$	Saturated vapor energy for material M.
SIEGTB	PASS		Storage array for the previous row's advance-time vapor-specific internal energy.
SIEGU	RXCHA	$\tilde{e}_G^{n+1}$	Estimated vapor-mixture specific internal energy for the time t^{n+1} , given by Eq. (IV-251).
SIEL	SIMCOM (KOM)	e_{Lm}	The specific internal energy of the liquid-field energy component for a mesh cell.
SIELCR	SIMCOM (KOM)		The internal energy of a liquid material at an absolute temperature of two-thirds of the critical temperature.
SIELI	NITIAL	e_{Lm}^i	The initial specific internal energy of the liquid-field energy component for a mesh cell.
SIELIQ	EOS	$e_{\text{Liq},M}$	The internal energy of the liquid-field energy component at the melting temperature; that is, the liquidus energy.
SIELL	EOS	e_{Lm}	Temporary array to transfer a liquid component's internal energy to the EOS.
SIELS	EOS	$e_{\text{Con},M}$	The specific internal energy for the liquid-field component at the saturation temperature.
SIELN	RXCHA	$\tilde{e}_{Lm}^{n+1}$	Estimated advanced-time liquid-field energy component specific internal energy. For example, see Eq. (IV-72).

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
SIELQ	RXCHA		Internal energy per unit mass associated with melting mass transfers.
SIES	SIMCOM (KOM)	e_{Sm}	The specific internal energy of the structure-field energy component.
SIESST	RXCHA		Temporary storage of the control internal energy to be used if a mesh cell type is negative.
SIESEX	RXCHA		Internal energy per unit mass associated with freezing mass transfers.
SIESI	NITIAL	e_{Sm}^i	The initial specific internal energy of the structure-field energy component.
SIESL	EOS		The input internal energy for the structure-field materials.
SIESN	RXCHA	$\tilde{e}_{Sm}^{n+1}$	Estimated advanced-time specific internal energy of structure-field energy component m. For example, see Eq. (IV-73).
SIESOL	EOS	$e_{Sol,M}$	The specific internal energy of the structure-field components at the melting temperature (the solidus energy).
SIG	SIMCOM (KOM)	σ	The material surface tension.
SIGO	SIMCOM (IA)	$\bar{\sigma}$	Most recent value of the eigenvalue dominance ratio for the outer (fission-source) iteration of the flux-shape solution.
SILTB	PASS		Storage array for the previous row's advance-time liquid field energy component specific internal energy.
SIMA	SIMCOM (IA)		An array equivalenced to the A array in neutronics common block SIMCOM.
SIMFIA	SIMCOM (IA)		Array equivalenced to the FIA array for restart dumps.

TABLE J-I (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
SKEGO	SIMCOM (KOM)		A summation, by components, of vapor-field kinetic energy transport across the boundaries of the system.
SKELO	SIMCOM (KOM)		A summation, by components, of liquid-field kinetic energy transport across the boundaries of the system.
SMALL	CONST		A value defined as machine zero. Currently set to 10^{-20} .
SMGO	SIMCOM (KOM)		A summation, by components, of vapor-field momentum transport across the boundaries of the system.
SMI	SIMCOM (KOM)		Initial total mass in the system.
SMLO	SIMCOM (KOM)		A summation, by components, of liquid-field momentum transport across the boundaries of the system.
SNUC	SIMCOM (KOM)		The total nuclear heating.
SPEGO	SIMCOM (KOM)		A summation, by components, of pressure-volume work done by the vapor at the boundaries of the system.
SPOW	SIMCOM (KOM)		Current power because of flux shape.
SPOWS	SIMCOM (KOM)		Power resulting from flux shape at beginning of flux-shape time step.
SROG	SIMCOM (KOM)		A summation variable indicating whether a given vapor density component is present.
SROL	SIMCOM (KOM)		A summation variable indicating whether a given liquid density component is present.
SSGT	SIMCOM (KOM)	Γ_G	The vapor-field total mass-exchange rate.
SSLT	SIMCOM (KOM)	Γ_L	The liquid-field total mass-exchange rate.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ST00	STAT		Accumulated machine time spent in overlay 0,0.
ST1LAS	FWBGN1		Last entry in the FWBGN1 common block.
ST20	STAT		Accumulated machine time spent in overlay 2,0.
ST21	STAT		Accumulated machine time spent in overlay 2,1.
ST22	STAT		Accumulated machine time spent in overlay 2,2.
ST24	STAT		Accumulated machine time spent in overlay 2,4.
ST2LAS	FWBGN2		Last entry in the FWBGN2 common block.
ST30	STAT		Accumulated machine time spent in overlay 3,0.
ST31	STAT		Accumulated machine time spent in overlay 3,1.
ST32	STAT		Accumulated machine time spent in overlay 3,2.
ST33	STAT		Accumulated machine time spent in overlay 3,3.
ST34	STAT		Accumulated machine time spent in overlay 3,4.
ST35	STAT		Accumulated machine time spent in overlay 3,5.
ST36	STAT		Accumulated machine time spent in overlay 3,6.
ST37	STAT		Accumulated machine time spent in overlay 3,7.
ST38	STAT		Accumulated machine time spent in overlay 3,8.
ST39	STAT		Accumulated machine time spent in overlay 3,9.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
ST3A	STAT		Accumulated machine time spent in overlay 3,10.
SUM	RXCHA		Temporary variable used to sum various quantities.
SUM3	RXCHA		Variable summing energy changes in the multicomponent vaporization/condensation model.
SUM4	RXCHA		Variable summing energy changes per unit temperature in the multicomponent vaporization/condensation model.
ST3B	STAT		Accumulated machine time spent in overlay 3,11.
ST5LAS	LOCAL		Last entry in the LOCAL common block.
STCAR	RXCHA	A_{Gsk}	Effective structure area for the vaporization/condensation modeling.
STCHT	RXCHA	h_{Sk}	Structure-side heat-transfer coefficient in the simple vaporization/condensation model.
STRCWT	SIMCOM (KOM)	w_{SS}	Stainless-steel reactivity worth distribution.
STRO	TIME		The total mass of liquid fuel, steel, and sodium summed over the fluid dynamics mesh.
STROI	TIME		The total energy of liquid fuel, steel, and sodium summed over the fluid dynamics mesh.
STVRO	TIME		The cumulative sum of STRO for all fluid-dynamic time steps since the last neutronics step.
STVROI	TIME		The cumulative sum of STROI for all fluid-dynamic time steps since the last neutronics step.
STQGHT	RXCHA		Array used for direct structure-vapor energy transfer problem.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
SUM	RXCHA		Temporary variable used to sum various quantities.
SUM3	RXCHA		Variable summing energy changes in the multicomponent vaporization/condensation model.
SUM4	RXCHA		Variable summing energy changes per unit temperature in the multicomponent vaporization/condensation model.
SUM5	RXCHA		Variable summing heat capacity per unit volume in the multicomponent vaporization/condensation model.
SUMA	SIMCOM (IA)		Temporary variable.
SUMA	RXCHA	A	First coefficient in the quadratic Eq. (IV-114).
SUMB	SIMCOM (IA)		Temporary variable.
SUMB	RXCHA	B	Second coefficient in the quadratic Eq. (IV-114).
SUMETA	SIMCOM (IA)		$\sum_{m=1}^{MM} \text{COSETA}(m) * \text{WGT}(m).$
SUMFIJ	RXCHA		<p>Product of the area and the heat-transfer coefficient,</p> $\sum_{k=1}^2 A_{sk} (h_{GS} + h_{Sk}) ,$ <p>used in the denominator of Eq. (III-173) and Eq. (III-174) in partitioning drag heating.</p>
SUMFJR	RXCHA		The inverse of SUMFIJ.
SUMH	RXCHA		<p>Summation over heat-transfer coefficient/area products used in Eq. (IV-81).</p> $\sum_{k \neq 4} H_{S4k} .$

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
SUMHT	RXCHA		Same as SUMH with each term multiplied by the estimated end-of-time-step temperature, $\sum_{k \neq 4} H_{S4k} \tilde{T}_k^{n+1} .$
SUMKIJ	RXCHA		Product of the area and the heat-transfer coefficient used in the denominator of Eqs. (III-171) and (III-172) in partitioning drag heating, given by $\sum_{m=1}^{NCLE} (h_{GLm} + h_{LGm}) A_{Lm} .$
SUMMU	SIMCOM (IA)		$\sum_{m=1}^{MM} COSMU(M) * WGT(M) .$
SUMN	SIMCOM (IA)		The integral of the amplitude function over a reactivity time step.
SUMPHI	RXCHA	$\sum_{k=1}^{NMAT} w_k \phi_{mk}$	Denominator in Eqs. (III-85) and (III-86).
SUMT	RXCHA	$\tilde{\rho}_G^{n+1} - \tilde{\rho}_G^n$	Macroscopic vapor-field density difference. Used in determination of DTL(8).
SURFT	RXCHA		Value of the surface temperature in the multicomponent vaporization-condensation model.
SVEL	ALLSET	$c_{\alpha m}$	Sonic velocity for liquid field density component m.
T	SIMCOM (IA)		Cell-centered flux used in FIXUP.
T	TIME		$T = TKACH$ = most recent real time for the fluid dynamics calculation.
T2OMD	TIML		Restart dump interval.
TAUNUM	RXCHA		Heat capacity per unit volume for either refrozen fuel or control.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
TCRIT	SIMCOM (KOM)	$T_{\text{Crt},M}$	The material critical temperature.
TCYC	TIME		Machine time for the next restart dump.
TDAT	TIME		Date of starting current SIMMER-II
TDIFF	RXCHA		Temporary variable describing a constant part of the Newton-Raphson procedure for updating the vapor-side heat-transfer coefficient for a structure surface.
TDMP	TIME		The times at which DTDMP is changed, controlling the restart dump frequency.
TEMGAS	SIMCOM (KOM)	T_G	Vapor temperature. Used as an initial estimate and returned by the EOS.
TEMP	RXCHA		Fabricated fuel temperature assigned to fission gas upon its release from intergranular gas into the vapor field.
TEMP1	SIMCOM (IA)		Temporary storage.
TEMP2	SIMCOM (IA)		Temporary storage.
TEMP3	SIMCOM (IA)		Temporary storage.
TEMP10	SIMCOM (KOM)	$\bar{T}_{fo}$	Initial temperature of the fertile fuel as used to calculate the Doppler reactivity feedback for a point kinetics calculation.
TERM1	RXCHA		Temporary storage.
TERM2	RXCHA		Temporary variable used in advancing surface temperatures in the multicomponent phase-transition model.
TERM1T	SIMCOM (KOM)		Temporary array used in the pressure iteration to avoid recalculating terms for the iteration parameter.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
TERM2T	SIMCOM (KOM)		Temporary array used in the pressure iteration to avoid recalculating terms for the iteration parameter.
TG	EOS	T_G	The vapor temperature at time t .
TGC	RXCHA	T_G^n	The vapor temperature before it is updated from vaporization/condensation or other processes.
TGI	NITIAL	T_G^i	The initial vapor temperature.
TGN	RXCHA	T_G^κ	The vapor temperature after κ iterations in the simple vaporization/condensation model.
TGNP0	RXCHA	$T_G^{\kappa-1}$	TGN at $\kappa-1$ iterations.
TGNP20	RXCHA	$T_G^{\kappa-2}$	The vapor temperature, TGN, at $\kappa-2$ iterations.
THCANW	SIMCOM (KOM)	t_{can}^i	Thickness of the subassembly can wall.
THCNGT	RXCHA	k_G	Thermal conductivity of the vapor mixture. Given by Eq. (III-86).
THCONL	SIMCOM (KOM)	k_{Lm}	The thermal conductivity of a material in the liquid phase.
THCONS	SIMCOM (KOM)	k_{Sm}	The thermal conductivity of a material in the solid phase.
THI	SIMCOM (KOM)		Impulse to the head.
TI	SIMCOM (IA)		Temporary storage for $\Psi_{i\pm 1/2jmg}$.
TIMACC	TIME		Dummy variable to determine the problem execution time.
TIMLAS	TIME		The last entry in the TIME common block.
TIMLIM	TIME		Dummy variable used to estimate the problem execution time.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
TEBETA	SIMCOM (IA)	β	Total effective delayed neutron fraction.
TIMN	TIME		Dummy variable used to calculate the problem execution time.
TINFO	RXCHA		Initial vapor temperature for the inner temperature iteration in the multicomponent phase transition model.
TJ (IA)	SIMCOM		Temporary storage for $\Psi_{ij\pm 1/2mg}$.
TKACH	TIME		Most recent real time for the fluid dynamics calculation.
TKIJ	SIMCOM (KOM)	k_{GL}^{n-1}	The liquid-vapor momentum exchange function from the previous cycle.
TKOEPS	RXCHA	T_G/ϵ_m	Dimensionless temperature used as the independent variable in computation of the collision integral, $\Omega_{k,m}$.
TL	EOS	T_{Lm}	The output liquid-field energy component temperature from the EOS; generally, the beginning-of-time-step temperature.
TLI	NITIAL	T_{Lm}^i	The initial liquid-field energy component temperatures.
TLIM	TIME		The amount of machine time allowed for the current problem.
TLIQ	RXCHA	$\hat{T}_{Lm}$	Liquid temperature used in the simple vaporization/condensation model. Defined by Eqs. (IV-90) to (IV-92).
TM	SIMCOM (IA)		Temporary storage for $\Psi_{ijm\pm 1/2g}$.
TMBR1	SIMCOM (KOM)	T_{Fert}	Mass-averaged temperature of all fertile fuel components.
TMBR2	SIMCOM (KOM)	T_{Fiss}	Mass-averaged temperature of all fissile fuel components.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
TMBR3	SIMCOM (KOM)	T_{SS}	Mass-averaged temperature for all stainless-steel components.
TMBR4	SIMCOM (KOM)	T_{Na}	Mass-averaged temperature of all sodium components.
TMBR5	SIMCOM (KOM)	T_{Cont}	Mass-averaged temperature of all control components.
TMEAVE	SIMCOM (KOM)		Time of the peak average pressure on the head.
TMEMAX	SIMCOM (KOM)		Time of the peak pressure on the head.
TMLT	EOS	$T_{Melt,M}$	A material's melting temperature.
TMPIN	SIMCOM (KOM)		Sodium temperature for the table of sodium inlet temperature-time.
TN	RXCHA	$\bar{T}_m^{n+1}$	Estimated temperature returned for the solution of the implicit heat-transfer equations Eq. (IV-70).
TOP	SIMCOM (KOM)		TOP is zero if the top boundary is rigid. TOP is 1 if the top boundary is outflow.
TOUDMP	TIME		The next output time for a restart dump.
TOUT	SIMCOM (KOM)		The next output time for plots and prints.
TOUTP	SIMCOM (KOM)		The next output time for the postprocessor dump.
TOVLH	STIM		Fluid dynamics overlay load execution time.
TOVLN	STIM		Neutronics overlay load execution time.
TOVLFS	STIM		Variable passed between overlays to determine overlay load execution time.
TPBOT	SIMCOM (KOM)		The times specified for the input table of the bottom boundary pressure vs time.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
TPFN	RXCHA	n	Martinelli correlation parameter used in Eqs. (III-157) and (III-158).
TPOW	SIMCOM (IA)		Current power integrated over the spatial mesh.
TPTOP	SIMCOM (KOM)		The times specified for the input table of the top boundary pressure vs time.
TPWF2	SIMCOM (KOM)	2R	Two-phase weighting factor. Input as R. Multiplied by 2 before use. See Eq. (III-150).
TQGC	RXCHA		The relative change in this parameter, $\frac{1}{h_{g,M}} (T_{Sat,M} - T_G) ,$ is a convergence test in the simple vaporization-condensation model.
TRANS	RXCHA		Important mass and energy transfer array in the multicomponent vaporization-condensation model.
TRATE	RXCHA	dT_{S1}/dt	Estimate for the time derivative of the fabricated fuel temperature rise rate.
TRIGHT	SIMCOM (KOM)		The specified times at which the right-side boundary pressure is PRIGHT.
TS	EOS	T_{Sm}	The structure-field energy component temperature array returned from the EOS.
TSAT	EOS	$T_{Sat,M}$	The saturation temperature at time t from the EOS.
TSATC	SIMCOM (KOM)	$T_{Sat,4}$	The saturation temperature for control material following execution of the vaporization-condensation model.
TSATF	SIMCOM (KOM)	$T_{Sat,1}$	The final saturation temperature for fuel as evaluated in the vaporization-condensation model.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
TSATNA	SIMCOM (KOM)	$T_{\text{Sat},3}$	The final saturation temperature for sodium from the vaporization-condensation model.
TSATS	SIMCOM (KOM)	$T_{\text{Sat},2}$	The final saturation temperature for steel following execution of the vaporization-condensation model.
TSHED	STIM		Fluid dynamics edit execution time.
TSHYD	TIME		Fluid dynamics calculation execution time.
TSI	NITIAL	T_{Sm}^i	The initial structure-field energy component temperature array.
TSMFL	STIM		Fluid dynamics implicit execution time (IMFLUD).
TSNEUT	STIM		Neutronics calculation execution time.
TSNL	RXCHA	$\tilde{T}_{\text{S4},i-1,j}^{n+1}$	Temperature in the can wall set-over model obtained from Eq. (IV-81).
TSNR	RXCHA	$\tilde{T}_{\text{S4},i+1,j}^{n+1}$	Similar to TSNL for the right-hand temperature. Obtained from solving Eq. (IV-79).
TSPHS	STIM		Phase transition calculation execution time (XCHANJ).
TSPKD	STIM		Amplitude equation calculation execution time (PKDRIV).
TSR	RXCHA	$T_{\text{S4},i+1,j}^n$	Can-wall temperature in cell (i+1,j) for use in the expression for TSNR.
TSSF	SIMCOM (KOM)		The fabricated fuel temperature that must be exceeded before fission gas release begins.
TSSH P	STIM		Flux-shape calculation execution time (GRIND2).
TSSHU	STIM		Data swapping execution time (SHUFFL).
TST	RXCHA		Structure temperature used for vaporization-condensation calculations.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
TSTAR	SIMCOM (KOM)	t_M^*	The parameter in the vapor pressure expression.
TSTN	RXCHA		Temporary surface temperature for an iteration in the multicomponent phase transition model.
TSTT	RXCHA		Temporary storage for TSTN.
TSUP	EOS		The degrees of superheat input for a given material.
TSXFL	STIM		Fluid dynamics explicit calculation execution time (EXFLUD).
TSXS	STIM		Cross-section calculation execution time (SHLDXS).
TTMPIN	SIMCOM (KOM)		The times specified for the input table of the liquid sodium inlet temperature vs time.
TTSAT	RXCHA	$T_{Sat,M}^{\kappa}$	Saturation temperature at iteration κ for the simple vaporization-condensation model.
TTSAT2	RXCHA	$T_{Sat,M}^{\kappa-2}$	Saturation temperature at iteration $\kappa-2$ for the simple vaporization-condensation model.
TTSATP	RXCHA	$T_{Sat,M}^{\kappa-1}$	Saturation temperature at iteration $\kappa-1$ for the simple vaporization-condensation model.
TVDIFF	RXCHA		Temporary variable describing a constant part of the Newton-Raphson procedure for updating the vapor-side heat-transfer coefficient for a droplet surface.
TVN	SIMCOM (KOM)	$\tilde{T}_G^{n+1}$	The final vapor-field temperature from the simple vaporization-condensation model.
TWFIN	TIME		Problem finish time (real time).
TXX	DUM		Current fluid dynamics real time passed to HTSPK.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
UG	SIMCOM (KOM)	U_G	The velocity of the vapor-field at the right boundary of a mesh cell.
UL	SIMCOM (KOM)	U_L	The velocity of the liquid-field at the right boundary of a mesh cell.
UNLAS	UNITS		Last entry in the UNITS common block.
VELFAC	SIMCOM (IA)	f	Neutron velocity factor for transient neutronics initialization.
VELG	RXCHA	$ \vec{v}_G $	Magnitude of the vapor velocity for a cell at the beginning of a time step.
VELL	RXCHA	$ \vec{v}_L $	Magnitude of the liquid velocity for a cell at the beginning of a time step.
VELSOD	SIMCOM (KOM)	C_{L1}	Velocity of sound for liquid fuel based on the local liquid fuel temperature.
VFCAN	RXCHA	$\bar{\rho}_{s6}/\rho_{s6}$	Volume fraction of can wall in a mesh cell.
VFCANI	RXCHA	α_{can}^i	Temporary variable set to VFCANW during execution of the structure configuration model.
VFCANW	SIMCOM (KOM)	α_{can}^i	Initial volume fraction of can wall for a region.
VFCLAD	SIMCOM (KOM)	α_{clad}^i	Initial volume fraction of cladding for a region.
VFCLD	RXCHA	$\bar{\rho}_{s5}/\rho_{s5}$	Volume fraction of cladding in a mesh cell.
VFCLDI	RXCHA	α_{clad}^i	Temporary variable set to VFCLAD during execution of the structure configuration model.
VFPELL	SIMCOM (KOM)	α_{pell}^i	Initial volume fraction of fuel pellets for a region.
VFPIN	RXCHA	α_p	Pin structure volume fraction.
VFPINI	RXCHA	α_p^i	Initial pin structure volume fraction. Defined following Eq. (III-99).

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
VFROZ	RXCHA	α_{froz}	Frozen structure volume fraction.
VG	SIMCOM (KOM)	V_G	The velocity of the vapor field at the top boundary of a mesh cell.
VL	SIMCOM (KOM)	V_L	The velocity of the liquid field at the top boundary of a mesh cell.
VLGFLM	TIME		Cumulative maximum vapor-field displacement since the previous post-processor dump.
VLGMOV	TIME		Maximum-allowed cumulative vapor field displacement between post-processor dumps.
VLLFLM	TIME		Cumulative maximum liquid-field displacement since the previous post-processor dump.
VLLMOV	TIME		Maximum-allowed cumulative liquid-field displacement between postprocessor dumps.
VLMG	RXCHA	$ v_{GL}^n $	The magnitude of the vapor-liquid velocity difference at the beginning of the time step.
VLMGN2	RXCHA	v_{GL}^2	The square of the vapor-liquid velocity difference used during the iteration to compute RP1.
VLMGNS	SIMCOM (KOM)	$\bar{v}_{GL}^{n+1}$	The difference between the liquid and vapor velocity used in the particle radius calculation.
VOLFX	RXCHA	α_{Lm}	The volume fraction of liquid energy component m.
VOLMIN	SIMCOM (KOM)		The minimum cell volume used in calculating RP4.
VP1	RXCHA	$ v_{GL}^n + P_1$	The first two terms on the right-hand side of Eq. (IV-60).
VPLUG	SIMCOM (KOM)		Velocity of the movable plug.
VREAT	KINET	$\Delta k/k$	Values of net reactivity.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
WALLP	SIMCOM (KOM)		Pressure at the vessel wall at the head.
WEBCRT	SIMCOM (KOM)	We	Value of the critical Weber number.
WTCOLR	PASS	$W_{Na,r}$	Radial shape of the sodium reactivity worth distribution.
WTCOLZ	PASS	$W_{Na,z}$	Axial shape of the sodium reactivity worth.
WTDOPR	PASS	DW_r	Radial shape of the Doppler weight distribution.
WTDOPZ	PASS	DW_z	Axial shape of the Doppler weight distribution.
WEBCRT	SIMCOM (KOM)	We_m	The critical Weber number.
WTFERR	PASS	$W_{Fert,r}$	Radial shape of the fertile fuel reactivity worth distribution.
WTFERZ	PASS	$W_{Fert,z}$	Axial shape of the fertile fuel reactivity worth distribution.
WTFISR	PASS	$W_{Fiss,r}$	Radial shape of the fissile fuel reactivity worth distribution.
WTFISZ	PASS	$W_{Fiss,z}$	Axial shape of the fissile fuel reactivity worth distribution.
WTMOL	SIMCOM (KOM)	A_m	The array of material molecular weights.
WTSTRR	PASS	$W_{SS,R}$	Radial shape of the stainless-steel reactivity worth distribution.
WTSRZ	PASS	$W_{SS,z}$	Axial shape of the stainless-steel reactivity worth distribution.
X	EOS	x_m	The vapor density mass fractions used by the EOS.
XCONV	SIMCOM (KOM)		X-coordinate conversion factor for transfer from the physical mesh coordinates to the 4020 frame coordinates.

TABLE J-I (CONT)

<u>FORTTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
XFLR	SIMCOM (KOM)	X_{Fm}	The fraction of failed structure material that is liquid.
XL	SIMCOM (KOM)		The left edge of the mesh for plots.
XL1	RXCHA	$X_{\ell 1 L1}$	Fraction of liquid fuel that is fertile.
XL2	RXCHA	$X_{\ell 2 L1}$	Fraction of liquid fuel that is fissile.
XMOLE	RXCHA	w_k	Mole fraction of vapor species k.
XMOV	TIME		Equivalence array for comparison of post-processor dump input controls with calculated values of the controls.
XMUG	SIMCOM (KOM)	μ_{gm}	The array of viscosities for the vapor-field components.
XMUGT	RXCHA	μ_G	The value for the vapor field viscosity.
XMUL	SIMCOM (KOM)	μ_{lm}	The array of input viscosities for the liquid-field components.
XNUM	RXCHA		Temporary storage. Generally contains numerators for fractions in the multi-component phase transition model.
XR	SIMCOM (KOM)		The right edge of the mesh for plots.
XSQRD	RXCHA	X^2	Martinelli parameter defined by Eq. (III-156).
XSRQT	UNITS		Request table for the cross-section random file.
XVUS	PSEDIT		Input scale factor for the X-view position.
YB	SIMCOM (KOM)		The bottom edge of the mesh for plots.
YCONV	SIMCOM (KOM)		Plotting factor for converting axial dimensions from problem units to 4020 units.

TABLE J-I (CONT)

<u>FORTRAN Variable</u>	<u>Common Block</u>	<u>Mathematical Symbol</u>	<u>Description</u>
YLD SIG	SIMCOM (KOM)		Tabular yield stress points corresponding to the temperatures given by YLDTMP.
YLDTMP	SIMCOM (KOM)		Tabular temperature points for the yield stress interpolation.
YT	SIMCOM (KOM)		The top edge of the mesh for plots.
YTL	RXCHA		Array containing the input/output quantities along the vaporization/sublimation curve for the tabular EOS.
YVUS	PSEDIT (KOM)		Input scale factor for the Y-view position.
ZBOT	SIMCOM		Bottom of node occupied by plug relative to ZPLUG.
ZETA	SIMCOM (KOM)	ξ_M	The array of exponent parameters in the heat of vaporization equation.
Z	SIMCOM (IA)		Radial geometry function used in FIXUP.
ZIP1	RXCHA		Temporary array for computing the affect of fission heating in calculating exchange functions. Product of EN2 and DTAMP.
ZPLUG	SIMCOM (KOM)		Position of movable plug. Normalized to zero at the start of the problem.
ZTOP	SIMCOM (KOM)		Top of node occupied by plug relative to ZPLUG.
ZVUS	PSEDIT		Input scale factor for the Z-view position.
ZZ	SIMCOM (IA)		Radial geometry function used in FIXUP.

APPENDIX K

SIMMER-II CONSTANTS

Because SIMMER-II must run on several different computing systems, the constants for the program have been collected into a single common block called CONST. The values for these constants are defined in subroutine NPUT1A. Depending on the operating system used, the constants are defined in either single precision (when the preprocessor option DBL is off) or double precision (when DBL is 1). The values for the constants are given below in Table K-I.

TABLE K-I

SIMMER-II CONSTANTS

<u>FORTTRAN Variable</u>	<u>Number Value</u>	<u>FORTTRAN Variable</u>	<u>Number Value</u>
ZERO	0.0	EM2	0.01
ONE	1.0	EM3	10^{-3}
TWO	2.0	EM4	10^{-4}
THREE	3.0	EM5	10^{-5}
FOUR	4.0	EM6	10^{-6}
FIVE	5.0	EM7	10^{-7}
SIX	6.0	EM8	10^{-8}
EIGHT	8.0	EM10	10^{-10}
NINE	9.0	EM12	10^{-12}
EP1	10.0	EM20	10^{-20}
EP2	100.0	SMALL	10^{-20}
EP3	1000.0	EM30	10^{-30}
EP4	10^4	PIE	π
EP5	10^5	PIO2	$\pi/2$
EP9	10^9	TWOPI	2π
EP10	10^{10}	C458M8	$4.5832 \cdot 10^{-8}$
EP20	10^{20}	C275M8	$2.75 \cdot 10^{-8}$
FLARGE	10^{20}	SIGRAD	$5.56 \cdot 10^{-8}$
EP30	10^{30}	C887M7	$8.875 \cdot 10^{-7}$
EM1	0.1	C770M7	$7.7 \cdot 10^{-7}$

<u>FORTTRAN Variable</u>	<u>Number Value</u>	<u>FORTTRAN Variable</u>	<u>Number Value</u>
C663M7	$6.637 \cdot 10^{-7}$	CP0502	$5.0246 \cdot 10^{-2}$
C119M7	$1.1976 \cdot 10^{-7}$	C546M2	$5.46452 \cdot 10^{-2}$
C972M6	$9.725 \cdot 10^{-6}$	CP0625	$6.25 \cdot 10^{-2}$
C267M6	$2.671 \cdot 10^{-6}$	CP0655	$6.5595 \cdot 10^{-2}$
C801M5	10/124716	C831M2	$8.314 \cdot 10^{-2}$
C762M5	$7.62 \cdot 10^{-5}$	CP086	$8.6 \cdot 10^{-2}$
C230M5	$2.30 \cdot 10^{-5}$	CP105	0.105
C208M5	$2.0887 \cdot 10^{-5}$	CP125	1/8
C939M4	$9.39 \cdot 10^{-4}$	C106	1/6
C936M4	$9.362 \cdot 10^{-4}$	CP13	0.13
C871M4	$8.718 \cdot 10^{-4}$	CP1985	0.19854
C500M4	$5.0 \cdot 10^{-4}$	CP2	0.2
C955M3	$9.55 \cdot 10^{-3}$	CP2387	$3/(4\pi)$
C698M3	$6.987 \cdot 10^{-3}$	CP25	1/4
C500M3	$5.0 \cdot 10^{-3}$	CP3	0.3
C388M3	$3.881 \cdot 10^{-3}$	THIRD	1/3
C261M3	$2.61 \cdot 10^{-3}$	CP35	0.35
C207M3	$2.07368 \cdot 10^{-3}$	CP375	0.375
C190M3	$1.9066 \cdot 10^{-3}$	CP3776	0.37765
C138M3	$1.383 \cdot 10^{-3}$	CP4	0.4
CP0128	$1.282 \cdot 10^{-2}$	CP45	0.45
CP0143	$1.4366 \cdot 10^{-2}$	HALF	1/2
CP0178	$1.789 \cdot 10^{-2}$	CP664	0.66448
CP0187	$1.8768 \cdot 10^{-2}$	TWO3RD	2/3
CP0217	$2.174 \cdot 10^{-2}$	CP7192	0.719288
CP0230	$2.3078 \cdot 10^{-2}$	CP75	0.75
CP025	$2.5 \cdot 10^{-2}$	CP8	0.8
CP0397	$3.97887357 \cdot 10^{-2}$	CP825	0.82515
CP0414	$4.144 \cdot 10^{-2}$	CP506	5/6
CP05	0.05	CP85	0.85
CP8888	0.88885	C2P5	2.5
CP9249	0.92495	C3P25	3.25
CP9	0.9	C2P75	2.75
CP95	0.95	C3P329	3.329
CP9700	0.97006	C3P707	3.7078

<u>FORTTRAN Variable</u>	<u>Number Value</u>	<u>FORTTRAN Variable</u>	<u>Number Value</u>
CP99	0.99	C3P75	3.75
CP999	0.999	C43P7	4.37
CP9995	0.9995	C4P47	4.47
CP9999	0.9999	C5P63	5.63
C1P067	1.0676	C6P67	6 2/3
C1P1	1.1	C15P	15.0
C1P135	1.13556	C16P	16.0
C1P15	1.15	C18P	18.0
C1P151	1.15049	C19P	19.0
CP3539	0.3539176	C20P	20.0
C1P25	1.25	C21P9	21.9
C1P32	1.32	C24P	24.0
C403	4/3	C50P	50.0
C1P42	1.42	C60P	60.0
C1P478	1.478	C69P31	69.3147
C1P5	1.5	C70P	70.0
C503	5/3	C87P4	87.4
SQRT3	$(3)^{1/2}$	C101P	101.0
C1P75	1.75	C119P	119.0
C2P25	2.25	C125P	125.0
C2P3	2.3	C131P	131.0
C2P303	LOG10(10)	C200P	200.0
C201P	201.0	C2000P	2000.0
C273P	273.0	C3000P	3000.0
C273P1	273.16	C101E5	$1.013 \cdot 10^5$
C300P	300.0	C300E5	$3.000 \cdot 10^5$
C301P	301.0	C192E6	$10^7/3(3)^{1/2}$
C401P	401.0	C250E5	$2.5 \cdot 10^5$
C450P	450.0	C101E6	$1.01325 \cdot 10^6$
C500P	500.0	C500E6	$5.0 \cdot 10^6$
C511P	511.0	C860E6	$8.6 \cdot 10^6$
C916P	916.0	C2212P	2212.339
C1022P	1022.0	C400P	400.0
C1200P	1200.0	C60P26	$6.0225 \cdot 10^{26}$

APPENDIX L

DERIVATION AND CALCULATIVE APPROACH FOR THE MULTIGROUP NEUTRON TRANSPORT EQUATIONS

The distribution in space, energy, and time of the neutrons in a nuclear reactor is a primary determinant of its behavior. The prediction of this distribution is accomplished by solving the neutron-transport equation.¹ The general form of the neutron transport equation formulates the neutron distribution in a seven-variable phase space (three spatial-coordinate variables, two directional-motion variables, one energy or speed variable, and one temporal variable). To obtain solutions to the transport equation in an expedient manner, several simplifying assumptions are made. The form of the transport equation given in Chap. III [Eq. (III-256)] already contains some of these assumptions. As mentioned in Chap. III, Eq. (III-256) contains the simplifying assumptions of limited anisotropic (transport-corrected P_0) scattering, no increase of neutron energy during scattering events, and symmetry in the θ coordinate of the r - z - θ cylindrical coordinate system. In this appendix, a derivation of the difference equations used in Chap. V is given, and the calculative sequence used by SIMMER-II is described. The bases for the assumptions made in arriving at the final form of the difference equations are indicated in this appendix, but Refs. 1-4 should be consulted for a detailed discussion. For convenience, the time-dependent, r - z neutron-transport equation and delayed neutron precursor equations are repeated.

$$\begin{aligned}
 & \frac{1}{V(E)} \frac{\partial \Psi}{\partial t} + \frac{\mu}{r} \frac{\partial(r\Psi)}{\partial r} - \frac{1}{r} \frac{\partial(z\Psi)}{\partial \omega} + \eta \frac{\partial \Psi}{\partial z} + \Sigma_t \Psi \\
 &= \frac{1}{4\pi} \int_E^\infty \Sigma_s(t, r, z, E' \rightarrow E) \Phi(t, r, z, E') dE' \\
 &+ \frac{\chi_p(E)}{4\pi k_0} \int_0^\infty \nu_p \Sigma_f(t, r, z, E') \Phi(t, r, z, E') dE' \\
 &+ \frac{1}{4\pi} \sum_{d=1}^{IGD} \lambda_d(E) \lambda_d C_d \quad , \quad (L-1a)
 \end{aligned}$$

$$\frac{\partial C_d}{\partial t} = \lambda_d C_d + \frac{1}{k_0} \int_0^\infty \nu_d \Sigma_f(t, r, z, E') \phi(t, r, z, E') dE' \quad d = 1, 2, \dots, \text{IGD} \quad ,$$

(L-1b)

where

$\Psi = \Psi(t, r, z, \vec{\Omega}, E)$ denotes the neutron flux per unit volume, per unit solid angle, and per unit energy at the spatial point (r, z) with direction $\vec{\Omega}$ and energy E at time t ;

$r, z, \mu, \eta, \xi, \omega$ are the spatial and angular coordinates for (r, z) geometry as shown in Fig. 6 (in Chap. 3). The relationship $\vec{\Omega} = \vec{\Omega}(\mu, \eta)$ between the directional cosines (μ, η, ξ) and the angle ω is given later;

$V(E)$ is the neutron velocity;

$\Sigma_t = \Sigma_t(t, r, z, E)$ is the macroscopic total cross section;

$\Sigma_s(t, r, z, E' \rightarrow E)$ denotes the macroscopic cross section for isotropic neutron scattering from neutron energy E' to neutron energy E ;

$\phi(t, r, z, E)$ denotes the neutron scalar flux;

k_0 denotes the initial stationary fission-source eigenvalue (k -effective) for the reactor system;

$\chi_p(E)$ is the prompt neutron fission emission spectrum;

ν_p is the prompt neutron fission yield;

$\Sigma_f(t, r, z, E)$ is the macroscopic fission cross section;

IGD is the number of delayed neutron precursor groups;

$\chi_d(E)$ is the delayed neutron emission spectrum from delayed neutron precursor group d ;

λ_d is the decay constant for delayed neutron precursor group d; and

$C_d = C_d(t, r, z)$ is the delayed neutron precursor concentration for precursor group d.

Equation (L-1a) is transformed by a variable change such that angular integrals yield unity instead of 4π ; that is,

$$\vec{\Omega}' = \frac{1}{4\pi} \vec{\Omega} \quad , \quad \text{and}$$

$$\Psi'(\vec{\Omega}') = 4\pi \Psi(\vec{\Omega}) \quad .$$

Equation (L-1a) becomes

$$\begin{aligned} \frac{1}{V(E)} \frac{\partial \Psi'}{\partial t} + \frac{\mu}{r} \frac{\partial(r\Psi')}{\partial r} - \frac{1}{r} \frac{\partial(\xi\Psi')}{\partial \omega} + \eta \frac{\partial \Psi'}{\partial z} + \Sigma_t \Psi' \\ = \int_E^\infty \Sigma_s(t, r, z, E' \rightarrow E) \Phi(t, r, z, E') dE' \\ + \frac{\chi_p(E)}{k_0} \int_0^\infty \nu \Sigma_f(t, r, z, E') \Phi(t, r, z, E') dE' \\ + \sum_{d=1}^{IGD} \chi_d(E) \lambda_d C_d \quad . \end{aligned} \quad (L-2)$$

The scalar flux is obtained by integrating the angular flux over all directions:

$$\begin{aligned} \Phi(t, r, z, E) &= \int_{4\pi} \Psi(t, r, z, \vec{\Omega}, E) d\Omega \\ &= \int_{4\pi} \Psi'(t, r, z, \vec{\Omega}', E) d\Omega' \quad . \end{aligned}$$

The "primed" notation is now omitted for simplicity.

I. DERIVATION OF THE MULTIGROUP EQUATIONS

One of the first steps normally used to obtain a solution of Eq. (L-2) (diffusion theory, discrete ordinates, and so on) is to make the multigroup approximation. In the method, the continuous energy variable is replaced by a finite number of discrete energy groups. The energy range of interest ($E_{\max} \geq E \geq E_{\min}$) is segmented into IGM groups by the energies E_g , where $g = 1, 2, \dots, \text{IGM}$. The ordering is such that the index g increases with decreasing energy (that is, $E_{g-1} > E_g$). To form the multigroup system of equations, Eq. (L-2) is integrated by

$$\int_{E_g}^{E_{g-1}} (\cdot) dE = \int_{\Delta E_g} (\cdot) dE \quad .$$

The result is

$$\begin{aligned} \frac{1}{v_g} \frac{\partial \psi_g}{\partial t} + \frac{\mu}{r} \frac{\partial(r\psi_g)}{\partial r} - \frac{1}{r} \frac{\partial(\xi\psi_g)}{\partial \omega} + \eta \frac{\partial \psi_g}{\partial z} + \Sigma_{t,g} \psi_g = \sum_{g'=1}^g \Sigma_{s,g' \rightarrow g} \phi_{g'} \\ + \frac{\chi_{pg}}{k_0} \sum_{g'=1}^{\text{IGM}} (\nu_p \Sigma_f)_{g'} \phi_{g'} + \sum_{d=1}^{\text{IGD}} \chi_{dg} \lambda_d C_d \quad , \end{aligned} \quad (\text{L-3})$$

where

$$\psi_g = \int_{\Delta E_g} \Psi(t, r, z, \vec{\Omega}, E) dE \quad ,$$

$$\phi_g = \int_{\Delta E_g} \Phi(t, r, z, E) dE \quad ,$$

hence ψ_g and ϕ_g are the total angular and scalar flux in the interval ΔE_g . The other quantities in Eq. (L-3) with the "g" subscript (known as group constants) are flux-weighted averages over the ΔE_g range. For example,

$$\Sigma_{t,g} = \frac{1}{\psi_g} \int_{\Delta E_g} \Sigma_t(t, r, z, E) \Psi(t, r, z, \vec{\Omega}, E) dE \quad ,$$

preserves the total reaction rate in the ΔE_g energy range.

The set of multigroup equations defined above are formally exactly equivalent to Eq. (L-2). However, because in practice the actual energy dependence of flux within each group is not known, an approximation is involved in the multigroup method as a result of approximating this energy dependence and the subsequent evaluation of the group constants. The number of energy groups and their segmenting energies used in a problem should be related to the accuracy of the flux estimate and the energy dependence of the cross sections. The science (or art) of forming multigroup cross section sets has been extensively covered in the literature and is the function of many sophisticated computer codes.

II. DERIVATION OF THE DISCRETE ORDINATES EQUATIONS

The basis of the S_N or discrete ordinates method of obtaining solutions to the neutron transport equation is to evaluate the angular distribution of the neutron flux in a number of discrete directions.^{1,2} A discrete direction $\vec{\Omega}_m$ can be specified by the directional cosine coordinates (μ_m, η_m, ξ_m) along the r, z , and θ axes. Figure 6 (Chap. III) illustrates the relationship between the spatial coordinate systems (r, z, θ) and the directional coordinate system (μ, η, ξ) . By definition, the discrete directional cosines (and also the continuous cosines) must satisfy the relationship $\mu_m^2 + \eta_m^2 + \xi_m^2 = 1$, as a consequence of the fact that there are only two independent angular variables. In r - z geometry the two independent angular variables are taken to be (μ_m, η_m) or (η_m, ω_m) . As illustrated in Fig. 6 (Chap. III), ω is the angle of neutron flight out of the r - z plane. The interrelationships are specified by

$$\mu_m = (1 - \eta^2)^{1/2} \cos \omega_m \quad \text{and}$$

$$\xi_m = (1 - \eta^2)^{1/2} \sin \omega_m \quad . \quad (L-4)$$

The term in Eq. (L-3) containing ω , that is, $(\frac{1}{r} \frac{\partial(\xi\Psi)}{\partial\omega})$, is the angular derivative, or angular redistribution, term. In curvilinear geometries (for example, cylindrical), the orientation of the angular coordinate changes as the spatial position of the neutron changes. The angular redistribution term accounts for the neutron's directional transfer as it streams. One characteristic of this term that will be used later is that its integral over all angle (that is, $\int_{4\pi} d\vec{\Omega}$) is zero.

As a first step in making the discrete ordinates approximation, the integral for the scalar flux is evaluated by a numerical quadrature formula such that

$$\phi_g(t, r, z) = \int_{4\pi} \psi_g(t, r, z, \vec{\Omega}) d\vec{\Omega} = \sum_{m=1}^{4*M} w_m \psi_{mg}(t, r, z) \quad (L-5)$$

where M is the number of directions in one octant of the unit sphere and each m is associated with a direction domain. Because r - z problems are by definition symmetric in the ξ direction [that is, $\psi(+\xi) = \psi(-\xi)$], only the $4 * M$ quadrature points in the $-\xi$ hemisphere are used. The w_m 's are the quadrature weights or weighting factors, which satisfy the relationship

$$\sum_{m=1}^{4*M} w_m = \int_{4\pi} d\Omega = 1.0 \quad .$$

The quadrature set formed by the $4 * M$ sets of $(w_m, \mu_m, \eta_m, \xi_m)$ can be viewed as a set of points on the unit sphere with coordinates (μ_m, η_m, ξ_m) and associated areas (w_m). The ordering of direction cosines is shown in Figs. L-1 and L-2 for S_4 and S_8 , respectively. The numbers in parenthesis in Fig. L-1 indicate the actual numbering of the of the direction cosines; the other numbers are the order of progression through the angular mesh in the calculation. In addition to the two constraints mentioned above,

$$(\mu^2 + \eta^2 + \xi^2 = 1.0 \text{ and } \sum_{m=1}^{4*M} w_m = 1.0) \quad ,$$

S_N quadrature sets generally satisfy some symmetry requirements.^{2,3} These requirements preserve some degree of rotational invariance and aid in the evaluation of the angular redistribution term. Any remaining degrees of freedom in selecting the quadrature set parameters generally are used to satisfy angular moment relationships^{3,4} such as

$$\sum_{m=1}^M w_m \mu_m^k \eta_m^l = \frac{2}{\pi} \int_0^1 \int_0^{\pi/2} \mu^k \eta^l d\omega d\eta \quad .$$

Fig. L-1.
The ordering of S_4 directions.

Fig. L-2.
The ordering of S_8 direction.

Procedures such as this are used to obtain reasonably accurate, general-purpose quadrature sets.⁴

The transport equation for particles streaming in the discrete direction specified by the r-z directional cosines (μ_m, η_m) can be written as

$$\frac{1}{V_g} \frac{\partial \Psi_{mg}}{\partial t} + \frac{\mu_m}{r} \frac{\partial (r \Psi_{mg})}{\partial r} - \frac{1}{r} \frac{\partial (\xi \Psi_g)}{\partial \omega} \Big|_m + \eta_m \frac{\partial \Psi_{mg}}{\partial z} + \Sigma_{t,mg} \Psi_{mg} = S'_g, \quad (L-6)$$

where Ψ_{mg} is implicitly defined by Eq. (L-5), and

$$S'_g = \sum_{g'=1}^g \Sigma_{s,g' \rightarrow g} \phi_{g'} + \frac{\chi_{pg}}{k_0} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{g'} \phi_{g'} + \sum_{d=1}^{IGD} x_{dg} \lambda_d C_d.$$

With the incorporation of the discrete ordinates approximation, the angular redistribution term [third term in Eq. (L-6)] is placed in a compatible form. A finite-difference approximation of the angular derivative term will take the form

$$\frac{\partial (\xi \Psi_g)}{\partial \omega} \Big|_m \approx - \frac{\alpha_{m+1/2} \Psi_{m+1/2g} - \alpha_{m-1/2} \Psi_{m-1/2g}}{w_m}, \quad (L-7)$$

where the $m \pm 1/2$ index denotes the edges or surfaces of a directional cell about the direction m . To establish a recursion relationship for the α 's, consider the angular derivative when the flux is isotropic:

$$\frac{\partial (\xi \Psi_g)}{\partial \omega} = \frac{\partial \xi}{\partial \omega} \Psi_g + \xi \frac{\partial \Psi_g}{\partial \omega} = \frac{\partial \xi}{\partial \omega} \Psi_g = \mu \Psi_g. \quad (L-8)$$

If

$$\alpha_{m+1/2} - \alpha_{m-1/2} = -w_m \mu_m, \quad (L-9)$$

then, in the limit of small differences, the finite-difference form, and the analytic form give the same result for isotropic fluxes. Because neutron streaming changes only the (μ, ξ) directional cosines and the η cosine remains constant, it is appropriate to define the recursion of Eq. (L-8) along lines or arcs of constant η . Figures L-1 and L-2 show the projection of a typical quadrature set onto the μ - η plane. The angular differencing is performed along the dotted, constant η lines. Another restriction on the α coefficients is that the redistribution process should not create or destroy neutrons. When the difference expression is multiplied by w_m and summed over all directions (integrated over ω):

$$\sum_{m=1}^M (\alpha_{m+1/2} \psi_{m+1/2g} - \alpha_{m-1/2} \psi_{m-1/2g}) = \alpha_{M+1/2} \psi_{M+1/2g} - \alpha_{1/2} \psi_{1/2g} = 0.$$

If the quadrature set is ordered so that neutrons stream out of the first directional cell, then $\alpha_{1/2} = 0$. Likewise, if neutrons flow into, but not out of, the last directional cell, then $\alpha_{M+1/2} = 0$ and the finite-difference approximation for the angular redistribution term is conservative. If $\alpha_{1/2} = 0$, then the quadrature set must satisfy the relationship

$$\sum_{m=1}^M w_m \mu_m = 0$$

in order for $\alpha_{M+1/2} = 0$. This relationship is satisfied for quadrature sets that are symmetric in μ , about $\mu = 0$.

The discrete ordinates form of the transport equation, Eq. (L-6), is now written as

$$\frac{1}{v_g} \frac{\partial \psi_{mg}}{\partial t} + \frac{\mu_m}{r} \frac{\partial (r \psi_{mg})}{\partial r} + \frac{1}{w_m r} (\alpha_{m+1/2} \psi_{m+1/2g} - \alpha_{m-1/2} \psi_{m-1/2g})$$

$$+ \eta_m \frac{\partial \psi_{mg}}{\partial z} + \Sigma_{t,mg} \psi_{mg} = S'_g \quad (L-10)$$

using the terms as defined by Eq. (L-6).

III. DERIVATION OF SPATIAL DIFFERENCE EQUATIONS

The spatial domain is partitioned into $I_{\max}$ intervals in the r -direction and $J_{\max}$ intervals in the z -direction. Spatial cell boundaries are indicated with a $\pm 1/2$ subscript, with i denoting the r interval subscript and j denoting the z interval subscript. Cell (ij) has a center at r_i, z_j , and boundaries at $r = r_{i+1/2}$, and $r_{i-1/2}$, and the z boundaries at $z = z_{j+1/2}$ and $z_{j-1/2}$. Derivation of the r - z finite-difference equations will require definitions for cell volume and surface averages. The cell volume integral is

$$V_{ij} = \int_{r_{i-1/2}}^{r_{i+1/2}} 2\pi r dr \int_{z_{j-1/2}}^{z_{j+1/2}} dz = \pi(r_{i+1/2}^2 - r_{i-1/2}^2)(z_{j+1/2} - z_{j-1/2}) \quad (L-11a)$$

and the cell average flux is defined as

$$\psi_{ijmg}(t) = \frac{\int_{r_{i-1/2}}^{r_{i+1/2}} 2\pi r dr \int_{z_{j-1/2}}^{z_{j+1/2}} \psi_{mg}(r, z, t) dz}{V_{ij}} \quad (L-11b)$$

The cell surface areas are

$$A_{i\pm 1/2j} = 2\pi r_{i\pm 1/2} \int_{z_{j-1/2}}^{z_{j+1/2}} dz = 2\pi r_{i\pm 1/2}(z_{j+1/2} - z_{j-1/2}) \quad (L-11c)$$

in the axial direction, and

$$B_{i+1/2} = B_{i-1/2} = B_i = 2\pi \int_{r_{i-1/2}}^{r_{i+1/2}} r dr = \pi(r_{i+1/2}^2 - r_{i-1/2}^2) \quad (L-11d)$$

in the radial direction. The surface average fluxes are

$$\psi_{i\pm 1/2jmg} = \frac{2\pi r_{i\pm 1/2} \int_{z_{j-1/2}}^{z_{j+1/2}} \psi_{mg}(t, r_{i\pm 1/2}, z_j) dz}{A_{i\pm 1/2}} \quad \text{and} \quad (L-11e)$$

$$\psi_{ij\pm 1/2mg} = \frac{\int_{r_{i-1/2}}^{r_{i+1/2}} 2\pi r \psi_{mg}(t, r_i, z_{j\pm 1/2}) dr}{B_i} \quad (L-11f)$$

Integrating Eq. (L-10) over the spatial cell volume and using the averages defined in Eq. (L-11) yields

$$\begin{aligned} \frac{V_{ij}}{V_g} \frac{\partial \psi_{ijmg}}{\partial t} + \mu_m (A_{i+1/2j} \psi_{i+1/2jmg} - A_{i-1/2j} \psi_{i-1/2jmg}) \\ + \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j}) (\alpha_{m+1/2} \psi_{ijm+1/2g} - \alpha_{m-1/2} \psi_{ijm-1/2g}) \\ + \eta_m B_i (\psi_{ij+1/2mg} - \psi_{ij-1/2mg}) + V_{ij} \Sigma_{t,ijg} \psi_{ijmg} = V_{ij} S'_{ijg} \quad (L-12) \end{aligned}$$

where the total cross section, $\Sigma_{t,mg}$, and the source term, S'_g , are converted to cell-averaged quantities, $\Sigma_{t,ijg}$ and S'_{ijg} , respectively, in the same manner as the flux, by means of Eq. (L-11b).

Equation (L-12) is a spatially finite-differenced equation for the time-dependent multigroup discrete-ordinates equations; however, the time derivative is not expressed in finite-difference form. In SIMMER-II the quasistatic method is used to treat the time dependence; it is explained in detail in Chap. V. However, the flux factorization, upon which the quasistatic method is based, is introduced to obtain the form for the time dependence that appears in the SIMMER-II flux-shape solution modules.

In the quasistatic method, the time-dependent angular flux is factored into an amplitude function, $N(t)$, and a flux shape, $\psi(t, r, z, \vec{\Omega}, E)$; that is,

$$\Psi(t, r, z, \vec{\Omega}, E) = N(t) \psi(t, r, z, \vec{\Omega}, E) \quad (L-13)$$

where N contains most of the time dependence and the flux shape is slowly varying in comparison. The scalar flux similarly is factored into the (same) amplitude function, $N(t)$, and a flux shape, $\phi(t,r,z,E)$. The partial-differential flux equation thus is split into two equations: an ordinary-differential equation set for N that is easily solved and the partial-differential flux-shape equation, which is solved only when significant changes in the flux shape have occurred.

Substitution of Eq. (L-13) in the first term of Eq. (L-12) yields

$$\frac{V_{ij}}{V_g} \frac{\partial \psi_{ijmg}}{\partial t} = N \frac{V_{ij}}{V_g} \frac{\partial \psi_{ijmg}}{\partial t} + \psi_{ijmg} \frac{V_{ij}}{V_g} \frac{dN}{dt} . \quad (L-14)$$

The partial derivative of the flux shape is replaced by a backward difference relation, that is,

$$\frac{\partial \psi_{ijmg}}{\partial t} \approx \frac{\psi_{ijmg}^k - \psi_{ijmg}^{k-1}}{\Delta t} , \quad (L-15)$$

where the $k-1$ superscript indicates the flux shape at $t-\Delta t$, that is, the previous time step. In the following, the explicit temporal dependence of ψ_{ijmg} is no longer needed because of Eq. (L-15); thus, for simplicity, the k superscript on ψ_{ijmg}^k will be dropped. Because $\frac{dN}{dt}$ is replaced by a numerical value determined outside the flux shape solution it is left in differential form. Substitution of Eqs. (L-13) and (L-15) into Eq. (L-12) yields, after division by N ,

$$\begin{aligned} & \mu_m (A_{i+1/2j} \psi_{i+1/2jmg} - A_{i-1/2j} \psi_{i-1/2jmg}) \\ & + \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j}) (\alpha_{m+1/2} \psi_{ijm+1/2g} - \alpha_{m-1/2} \psi_{ijm-1/2g}) \\ & + \eta_m B_i (\psi_{ij+1/2mg} - \psi_{ij-1/2mg}) + \Sigma'_{t,ijg} \psi_{ijmg} = S_{ijmg} , \end{aligned} \quad (L-16)$$

where

$$\Sigma'_{t,ijg} = V_{ij}(\Sigma_{t,ijg} + \frac{1}{V_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{V_g} \frac{1}{\Delta t})$$

$$S_{ijmg} = V_{ij} \left(\sum_{g'=1}^g \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} + \frac{x_{pg}}{k_o} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} \right. \\ \left. + \frac{1}{N} \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd} + \frac{1}{V_g} \frac{1}{\Delta t} \psi_{ijmg}^{k-1} \right) .$$

Equation (L-16) contains four unknowns, assuming that three values of ψ are available from boundary conditions or previous cell solutions. Equations describing the relationship between fluxes at cell boundaries are needed for solution. The "diamond" difference relations, which state that the average cell flux is given by an average of the cell boundary values, provide a simple recursion relation and give second-order accuracy. However, using the diamond formulation may produce negative fluxes, the treatment of which requires a nonlinear "fixup" procedure that in some cases may cause convergence problems. An alternative formulation is the "weighted diamond" difference relations, in which boundary fluxes are given unequal weights determined in such a way as to produce only zero or positive angular fluxes. Both the standard diamond and weighted diamond treatments are available as options in SIMMER-II.

A. "Diamond" Difference Solution

Three equations are created by using the diamond difference relations.

$$\psi_{ijmg} = \frac{1}{2} (\psi_{i+1/2jmg} + \psi_{i-1/2jmg}) \quad (L-17a)$$

$$= \frac{1}{2} (\psi_{ij+1/2mg} + \psi_{ij-1/2mg}) \quad (L-17b)$$

$$= \frac{1}{2} (\psi_{ijm+1/2g} + \psi_{ijm-1/2g}) \quad (L-17c)$$

For $\mu_m > 0$, $\psi_{i-1/2jmg}$ is known and $\psi_{i+1/2jmg}$ is unknown with the reverse true for $\mu_m < 0$. Likewise for $\eta_m < 0$, $\psi_{ij-1/2mg}$ is known with $\psi_{ij+1/2mg}$ to be determined.

$\psi_{ijm-1/2g}$ is always known, with $\psi_{ijm+1/2g}$ to be found. Equations (L-17) will be substituted into Eq. (L-16) in the following form.

$$\psi_{i\pm 1/2jmg} = 2\psi_{ijmg} - \psi_{i\mp 1/2jmg} \text{ for } \pm\mu_m,$$

$$\psi_{ij\pm 1/2mg} = 2\psi_{ijmg} - \psi_{ij\mp 1/2mg} \text{ for } \pm\eta_m, \text{ and}$$

$$\psi_{ijm+1/2g} = 2\psi_{ijmg} - \psi_{ijm-1/2g}.$$

The solution for ψ_{ijmg} is then

$$\psi_{ijmg} = \frac{N_D}{D_D}, \quad (L-18)$$

where

$$\begin{aligned} N_D &= |\pm\mu_m| (A_{i+1/2j} + A_{i-1/2j})\psi_{i\pm 1/2jmg} \\ &+ \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j})(\alpha_{m+1/2} + \alpha_{m-1/2})\psi_{ijm-1/2g} \\ &+ 2|\pm\eta_m| B_i \psi_{ij\mp 1/2mg} + S_{ijmg}, \text{ and} \\ D_D &= 2|\pm\mu_m| A_{i\pm 1/2j} + \frac{2}{w_m} (A_{i+1/2j} - A_{i-1/2j})\alpha_{m+1/2} + 2|\pm\eta_m| B_i + \sum'_{t,ijg} \\ &= 2|\pm\mu_m| A_{i\pm 1/2j} + \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j})\alpha_{m+1/2} \\ &+ \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j})\alpha_{m+1/2} + \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j})\alpha_{m-1/2} \\ &- \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j})\alpha_{m-1/2} + 2|\pm\eta_m| B_i + \sum'_{t,ijg} \\ &= 2|\pm\mu_m| A_{i\pm 1/2j} - |\pm\mu_m| (A_{i+1/2j} - A_{i-1/2j}) \\ &+ \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j})(\alpha_{m+1/2} - \alpha_{m-1/2}) + 2|\pm\eta_m| B_i + \sum'_{t,ijg} \end{aligned}$$

by use of Eq. (L-9), and using the definitions of Eq. (L-16). Finally,

$$D_D = |\pm\mu_m| (A_{i+1/2j} + A_{i-1/2j}) + 2 |\pm\eta_m| B_i \\ + \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j}) (\alpha_{m+1/2} + \alpha_{m-1/2}) + \sum'_{t,ijg} .$$

In Eq. (L-18), the purpose of including the sign of the μ_m and η_m within absolute value signs is to make clear which values of μ_m and η_m (positive or negative) are associated with which ψ 's and coefficient terms. To provide for a more compact notation and to facilitate a later explanation of the computational treatment, let

$$(AA)_{ijm} = |\pm\mu_m| (A_{i+1/2j} + A_{i-1/2j}) , \quad (L-19a)$$

$$(BB)_{im} = 2 |\pm\eta_m| B_i , \text{ and} \quad (L-19b)$$

$$(CC)_{ijm} = \frac{1}{w_m} (A_{i+1/2j} - A_{i-1/2j}) (\alpha_{m+1/2} + \alpha_{m-1/2}) . \quad (L-19c)$$

Additionally, let

$$\beta_{ijmg} = (AA)_{ijm} \psi_{i\mp 1/2jmg} + (BB)_{im} \psi_{ij\mp 1/2mg} \quad (L-20a)$$

$$+ (CC)_{ijm} \psi_{ijm-1/2g} + S_{ijmg}, \text{ and}$$

$$\gamma_{ijmg} = (AA)_{ijm} + (BB)_{im} + (CC)_{ijm} + \sum'_{t,ijg} . \quad (L-20b)$$

Equation (L-18) then can be written as

$$\psi_{ijmg} = \frac{\beta_{ijmg}}{\gamma_{ijmg}}, \text{ for } \pm\mu_m \text{ and } \pm\eta_m . \quad (L-21)$$

B. "Weighted Diamond" Difference Solution

In this approach, the angular differencing is the same as in the "standard diamond" approach; that is, repeating Eq. (L-17c),

$$\psi_{ijmg} = \frac{1}{2} (\psi_{ijm+1/2g} + \psi_{ijm-1/2g}) \quad .$$

Substitution of Eq. (L-17c) into Eq. (L-16) and using Eq. (L-9) leads to

$$\begin{aligned} & \mu_m (A_{i+1/2j} \psi_{i+1/2jmg} - A_{i-1/2j} \psi_{i-1/2jmg}) - \mu_m (A_{i+1/2j} - A_{i-1/2j}) \psi_{ijmg} \\ & + \eta_m B_i (\psi_{ij+1/2mg} - \psi_{ij-1/2mg}) \\ & + (CC)_{ijm} (\psi_{ijmg} - \psi_{ijm-1/2g}) + \sum'_{t,ijg} \psi_{ijmg} = S_{ijmg} \quad , \end{aligned} \quad (L-22)$$

where the definition equation [Eq. (L-19c)] has been used.

The weighted diamond relations for the spatial differencing are given by

$$(1+a) \psi_{i\pm 1/2jmg} + \psi_{i\mp 1/2jmg} = (2+a) \psi_{ijmg} \quad \text{for } \pm \mu_m \quad \text{and} \quad (L-23a)$$

$$(1+a) \psi_{ij\pm 1/2mg} + \psi_{ij\mp 1/2mg} = (2+a) \psi_{ijmg} \quad \text{for } \pm \eta_m \quad , \quad (L-23b)$$

where a is chosen to insure that $\psi_{i\pm 1/2jmg}$ and $\psi_{ij\pm 1/2mg}$ always are positive. Substitution of Eqs. (L-23) into Eq. (L-22) results in

$$\begin{aligned} & \frac{|\pm \mu_m|}{(1+a)} [(A_{i+1/2j} + A_{i-1/2j}) \psi_{ijmg} - (A_{i+1/2j} - A_{i-1/2j}) \psi_{i\mp 1/2jmg} \\ & + a A_{i\mp 1/2j} \psi_{ijmg} - a A_{i\mp 1/2j} \psi_{i\mp 1/2jmg}] \\ & + \frac{2|\pm \eta_m| B_i}{(1+a)} [\psi_{ijmg} - \psi_{ij\mp 1/2mg} + \frac{a}{2} \psi_{ijmg} - \frac{a}{2} \psi_{ij\mp 1/2mg}] \\ & + (CC)_{ijm} (\psi_{ijmg} - \psi_{ijm-1/2g}) + \sum'_{t,ijg} \psi_{ijmg} = S_{ijmg} \quad . \end{aligned} \quad (L-24)$$

Substitution of Eqs. (L-19a) and (L-19b) for $(AA)_{ijm}$ and $(BB)_{im}$, multiplication by $(1+a)$, simplification, and solution for ψ_{ijmg} yields

$$\psi_{ijmg} = \frac{N_w}{D_w} \quad , \quad (L-25)$$

where

$$\begin{aligned} N_w = & (AA)_{ijm} \psi_{i\mp 1/2jmg} + (BB)_{im} \psi_{ij\mp 1/2mg} + (CC)_{ijm} \psi_{ijm-1/2g} + S_{ijmg} \\ & + a (|\pm \mu_m| A_{i\mp 1/2j} \psi_{i\mp 1/2jmg} + \frac{1}{2} (BB)_{im} \psi_{ij\mp 1/2mg} \\ & + (CC)_{ijm} \psi_{ijm-1/2g} + S_{ijmg}) \quad \text{and} \\ D_w = & (AA)_{ijm} + (BB)_{im} + (CC)_{ijm} + \sum'_{t,ijg} + a (|\pm \mu_m| A_{i\mp 1/2j} + \frac{1}{2} (BB)_{im} \\ & + (CC)_{ijm} + \sum'_{t,ijg}) \quad . \end{aligned}$$

Rewritten in a condensed notation,

$$\psi_{ijmg} = \frac{[\beta_{ijmg} + a\alpha_{ijmg}]}{[\gamma_{ijmg} + a\delta_{ijmg}]} \quad , \quad (L-26)$$

where β_{ijmg} and γ_{ijmg} are defined as in Eqs. (L-20) and

$$\begin{aligned} \alpha_{ijmg} = & |\pm \mu_m| A_{i\mp 1/2j} \psi_{i\mp 1/2jmg} + \frac{(BB)_{im}}{2} \psi_{ij\mp 1/2mg} \\ & + (CC)_{ijm} \psi_{ijm-1/2g} + S_{ijmg} \quad \text{and} \end{aligned} \quad (L-27a)$$

$$\delta_{ijmg} = |\pm \mu_m| A_{i\mp 1/2j} + \frac{(BB)_{im}}{2} + (CC)_{ijm} + \sum'_{t,ijg} \quad . \quad (L-27b)$$

In Eq. (L-26), if $a = 0$, then the solution for ψ_{ijmg} is the diamond difference solution of Eq. (L-21); that is, $\psi_{ijmg} = \beta_{ijmg}/\gamma_{ijmg}$. If use of the diamond difference formula would produce a negative angular flux, this fact must be determined and a suitable value for a must be established. Let the maximum of the known boundary fluxes be $\psi_{\max}$, where $\psi_{\max} = \max(\psi_{i+1/2jmg}, \psi_{ij\mp mg})$. Combining Eqs. (L-23) and (L-26), a relation between the cell edge fluxes is obtained to determine the minimum of the unknown fluxes:

$$\psi_{\min} = \frac{(2+a) \left[\frac{\beta_{ijmg} + a\alpha_{ijmg}}{\gamma_{ijmg} + a\delta_{ijmg}} \right] - \psi_{\max}}{1+a} \quad (L-28)$$

If $a = 0$, then $\psi_{\min}$ will be positive if $2\beta_{ijmg} - \gamma_{ijmg}\psi_{\max} = \beta'_{ijmg}$ is positive. Therefore, calculatively, if $\beta'_{ijmg} \geq 0$, the standard diamond solution is positive and a is set to zero. If $\beta'_{ijmg} \leq 0$, then a will be set to a value that insures $\psi_{\min} \geq 0$. Assuming the numerator of Eq. (L-28) to be positive implies that

$$a^2 \alpha_{ijmg} + a(\beta_{ijmg} + 2\alpha_{ijmg} - \delta_{ijmg}\psi_{\max}) + \beta'_{ijmg} \geq 0 \quad (L-29)$$

Given that the incoming boundary fluxes are positive (which they should be with a positive differencing scheme), then α is always positive. With β'_{ijmg} being negative (by assumption), Eq. (L-29) will have two real roots, with at least one positive. If the minimum positive (real) root is used for a in Eqs. (L-23) and (L-26), the resulting fluxes will be nonnegative, closest to the diamond difference solution, and, hence, the most accurate for this type of a positive difference scheme.

However, finding the roots for Eq. (L-29) can be computationally expensive. To provide a less time-consuming scheme, it is assumed that $a = \beta'_{ijmg}/\alpha_{ijmg}$, and Eq. (L-29) is tested for positivity as follows.

$$\frac{\beta_{ijmg}^2}{\alpha_{ijmg}} - \frac{\beta'_{ijmg}}{\alpha_{ijmg}} (\beta_{ijmg} + 2\alpha_{ijmg} - \delta_{ijmg}\psi_{\max}) + \beta'_{ijmg} \geq 0 \quad .$$

Next, division by $-\frac{\beta'_{ijmg}}{\alpha_{ijmg}}$ (a positive number) yields

$$-\beta'_{ijmg} + \beta_{ijmg} + 2\alpha_{ijmg} - \delta_{ijmg} \psi_{\max} - \alpha_{ijmg} \geq 0 \quad .$$

Substituting for β' :

$$\gamma_{ijmg} \psi_{\max} + \alpha_{ijmg} - \beta_{ijmg} - \delta_{ijmg} \psi_{\max} \geq 0 \quad . \quad (L-30)$$

Substituting for β_{ijmg} , α_{ijmg} , γ_{ijmg} , and δ_{ijmg} yields

$$\begin{aligned} & ((AA)_{ijm} + (BB)_{im} + (CC)_{ijm} + \sum'_{t,ijm}) \psi_{\max} \\ & + (|\pm\mu_m| A_{i\mp 1/2j} \psi_{i\mp 1/2jmg} \\ & + \frac{(BB)_{im}}{2} \psi_{ij\mp 1/2mg} + (CC)_{ijm} \psi_{ijm-1/2g} + S_{ijmg}) \\ & - ((AA)_{ijmg} \psi_{i\mp 1/2jmg} + (BB)_{im} \psi_{ij\mp 1/2mg} + (CC)_{ijm} \psi_{ijm-1/2g} + S_{ijmg}) \\ & - (|\pm\mu_m| A_{i\mp 1/2j} + \frac{(BB)_{im}}{2} + (CC)_{ijm} + \sum'_{t,ijg}) \psi_{\max} \geq 0 \quad . \end{aligned}$$

Simplifying results in

$$\begin{aligned} & ((AA)_{ijm} - |\pm\mu_m| A_{i\mp 1/2j})(\psi_{\max} - \psi_{i\mp 1/2jmg}) \\ & + \frac{(BB)_{im}}{2} (\psi_{\max} - \psi_{ij\mp 1/2mg}) \geq 0 \quad . \end{aligned} \quad (L-31)$$

The flux differences in Eq. (L-31) are either zero or positive, and two other coefficients also are positive; thus, the assumption $a = \beta'_{ijmg}/\alpha_{ijmg}$ will yield a positive difference scheme. This of course assumes that the incoming boundary fluxes and sources are positive. It should be noted that the diamond

relationship has been used for the angular boundary fluxes and could lead to negative fluxes.

C. Solution Procedure

Equation (L-21) or (L-26) is used if three fluxes on the boundary of the phase space cell are known. To initialize the spatial boundary fluxes, Eqs. (L-21) or (L-26) is applied first at the outer mesh boundary for incoming directions. The initial $\psi_{ijm-1/2g}$ is approximated by a step function for the first incoming μ direction on each η level. This is done by setting $\psi_{ijm+1/2g} = \psi_{ijmg}$ and results in an equation like Eqs. (L-21) or (L-26) but with $(\alpha_{m+1/2} + \alpha_{m-1/2})$ equal to zero.

IV. DERIVATION OF THE DISCRETE ORDINATES FINE-MESH REBALANCE EQUATIONS

Equation (L-16) is solved iteratively because the RHS, that is, S_{ijmg} , depends on the angular flux. The iteration converges more rapidly if neutron conservation is enforced for each mesh cell.

Term-by-term summation of Eq. (L-16), after multiplication by w_m , over all angular cells yields the following.

Radial flow

$$\begin{aligned}
 & \sum_{m=1}^{4*M} w_m \mu_m (A_{i+1/2j} \psi_{i+1/2jmg} - A_{i-1/2j} \psi_{i-1/2jmg}) \\
 &= [A_{i+1/2j} \sum_{\mu_m > 0} w_m \mu_m \psi_{i+1/2jmg} - A_{i-1/2j} \sum_{\mu_m > 0} w_m \mu_m \psi_{i-1/2jmg}] \\
 &- [A_{i+1/2j} \sum_{\mu_m < 0} w_m \mu_m \psi_{i+1/2jmg} - A_{i-1/2j} \sum_{\mu_m < 0} w_m \mu_m \psi_{i-1/2jmg}] \\
 &= [FR_{i+1/2j} - FR_{i-1/2j}] - [FL_{i+1/2j} - FL_{i-1/2j}] \quad , \quad (L-32)
 \end{aligned}$$

which defines the right and left cell boundary flow terms. Flow toward the right is considered to be positive, and leftward flow is negative.

Angular flow

$$\sum_{m=1}^{4*M} (A_{i+1/2j} - A_{i-1/2j}) (\alpha_{m+1/2} \psi_{ijm+1/2g} - \alpha_{m-1/2} \psi_{ijm-1/2g}) = 0 ,$$

Axial flow

$$\begin{aligned} & \sum_{m=1}^{4*M} w_m \eta_m B_i (\psi_{ij+1/2mg} - \psi_{ij-1/2mg}) \\ &= [B_i \sum_{\eta_m > 0} w_m \eta_m \psi_{ij+1/2mg} - B_i \sum_{\eta_m > 0} w_m \eta_m \psi_{ij-1/2mg}] \\ & - [B_i \sum_{\eta_m < 0} w_m \eta_m \psi_{ij+1/2mg} - B_i \sum_{\eta_m < 0} w_m \eta_m \psi_{ij-1/2mg}] \\ &= [FU_{ij+1/2} - FU_{ij-1/2}] - [FD_{ij+1/2} - FD_{ij-1/2}] , \end{aligned} \quad (L-33)$$

which defines the upward and downward cell boundary flows. Upward flow is considered to be positive, and downward flow is negative.

Cell losses

$$\begin{aligned} & \sum_{m=1}^{4*M} w_m V_{ij} (\Sigma_{t,ijg} + \frac{1}{V_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{V_g} \frac{1}{\Delta t}) \psi_{ijmg} \\ &= V_{ij} (\Sigma_{t,ijg} + \frac{1}{V_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{V_g} \frac{1}{\Delta t}) \phi_{ijg} . \end{aligned} \quad (L-34)$$

Cell sources:

$$\sum_{m=1}^{4*M} w_m V_{ij} \left(\sum_{g'=1}^g \Sigma_{s,ijg'} \phi_{ijg'} + \frac{x_{pg}}{k_o} \sum_{g'=1}^{IGM} (v_p \Sigma_f)_{ijg'} \phi_{ijg'} \right)$$

$$\begin{aligned}
& + \frac{1}{N} \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd} + \frac{1}{V_g} \frac{1}{\Delta t} \psi_{ijmg}^{k-1}) \\
= & V_{ij} \left(\sum_{g'=1}^g \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} + \frac{x_{pg}}{k_o} \sum_{g'=1}^{IGM} (v_p \Sigma_f)_{ijg'} \phi_{ijg'} \right. \\
& \left. + \frac{1}{N} \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijd} + \frac{1}{V_g} \frac{1}{\Delta t} \phi_{ijg}^{k-1} \right) . \quad (L-35)
\end{aligned}$$

In the above terms, FU, FD, FR, and FL define the neutron flows from the upper, lower, right, and left surfaces of each mesh cell, respectively. Combining these terms with the loss and source terms does not yield a statement of neutron conservation unless the iteration over the within-group scattering source is converged fully. However, correction of the scalar fluxes to enforce cell-wise neutron conservation will accelerate convergence of the scattering iteration.

Rebalance factors, f_{ij} , are defined as

$$\phi_{ijg}^c = f_{ij} \phi_{ijg}^u , \quad (L-36)$$

where ϕ_{ij}^c is the corrected flux shape and ϕ_{ij}^u is the uncorrected flux shape. The rebalance factors are used to modify the flows in terms (L-32) through (L-34) before they are combined into a balance equation. However, the factors are not applied to the source term. An approximate correction is effected by assuming that the flows are directly proportional to the scalar flux in the cell from which the flow originates. The following relationships are substituted.

$$FR_{i+1/2j}^c = A_{i+1/2j} \sum_{\mu_m > 0} w_m \mu_m \psi_{i+1/2jmg}^c = f_{ij} FR_{i+1/2j}^u ,$$

$$FR_{i-1/2j}^c = A_{i-1/2j} \sum_{\mu_m > 0} w_m \mu_m \psi_{i-1/2jmg}^c = f_{i-1j} FR_{i-1/2j}^u ,$$

$$FL_{i+1/2j}^c = A_{i+1/2j} \sum_{\mu_m < 0} w_m \mu_m \psi_{i+1/2jmg}^c = f_{i+1j} FL_{i+1/2j}^u ,$$

$$FL_{i-1/2j}^c = A_{i-1/2j} \sum_{\mu_m < 0} w_m \mu_m \psi_{i-1/2jmg}^c = f_{ij} FL_{i-1/2j}^u ,$$

$$FU_{ij+1/2}^c = B_i \sum_{\eta_m > 0} w_m \eta_m \psi_{ij+1/2mg}^c = f_{ij} FU_{ij+1/2}^u ,$$

$$FU_{ij-1/2}^c = B_i \sum_{\eta_m > 0} w_m \eta_m \psi_{ij-1/2mg}^c = f_{ij-1j} FU_{ij-1/2}^u ,$$

$$FD_{ij+1/2}^c = B_i \sum_{\eta_m < 0} w_m \eta_m \psi_{ij+1/2mg}^c = f_{ij+1} FD_{ij+1/2}^u , \quad \text{and}$$

$$FD_{ij-1/2}^c = B_i \sum_{\eta_m < 0} w_m \eta_m \psi_{ij-1/2mg}^c = f_{ij} FD_{ij-1/2}^u .$$

Combining Eqs. (L-32) through (L-34) with the above substitutions yields a neutron conservation equation:

$$\begin{aligned} & - f_{i-1j} FR_{i-1/2j}^u - f_{i+1j} FL_{i+1/2j}^u - f_{ij-1} FU_{ij-1/2}^u - f_{ij+1} FD_{ij+1/2}^u \\ & + f_{ij} FR_{i+1/2j}^u + FL_{i-1/2j}^u + FU_{ij+1/2}^u + FD_{ij-1/2}^u \\ & + \left(\Sigma_{t,ijg} + \frac{1}{v_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{\Delta t} \frac{1}{v_g} \right) v_{ij} \phi_{ijg}] \\ & = (S_{ijg} + \frac{1}{\Delta t} \frac{1}{v_g} \phi_{ijg}^{k-1}) v_{ij} . \end{aligned} \tag{L-37}$$

Solution of Eq. (L-37) for the rebalance factors and modification of the scalar fluxes by use of Eqs. (L-36) yields a flux shape that satisfies the neutron conservation requirement. This process is performed once each scattering source iteration until the within-group scattering source has converged.

V. SIMMER-II CALCULATIVE EXPRESSIONS FOR THE DISCRETE ORDINATES EQUATIONS

The calculative approach used by SIMMER-II consists of a series of outer iterations that converge the prompt fission, downscatter, and delayed neutron sources. Within each outer iteration is a series of inner iterations for each group that converge the within-group scattering source. Each inner iteration consists of four sweeps applying Eq. (L-21) or Eq. (L-26): (1) the inward-downward sweep from the top and right boundaries $(-\mu_m, -\eta_m)$, (2) the outward-downward sweep from the top and left boundaries $(+\mu_m, -\eta_m)$, (3) the inward-upward sweep from the bottom and right boundaries $(-\mu_m, +\eta_m)$ and (4) the outward-upward sweep from the left and bottom boundaries $(+\mu_m, +\eta_m)$. Next, the scalar flux shapes are modified to satisfy cell-wise neutron conservation, as defined by the rebalance relations derived in the previous section.

In the following definitions, the references to SIMMER-II constants are intended to show the names of storage blocks. Information about the actual array indexing is not provided. Table L-I lists the definitions of the constants used in the SIMMER-II transport calculations. In the following equations, the quasistatic constraint variable γ (introduced in Chap. V) divides the fission source, as programmed in SIMMER-II. Although it is formally equal to unity (and thus does not affect the final solution), γ does vary during the quasistatic solution procedure in a way that accelerates convergence.

A. Calculative Expressions Based on "Diamond" Differencing

In the sweeps portion of the angular flux shape calculation, the terms representing the time dependence of the flux shape have been neglected because their inclusion requires storage of the angular flux shape between time steps. Because the flux shape is assumed to be slowly varying, this approximation has a negligible effect. Neglect of the time dependence of the flux shape involves dropping the ψ_{ijmg}^{k-1} term in both the total cross-section and source term, defined in Eq. (L-16). The term representing the time dependence of the amplitude function, $\frac{1}{v_g} \frac{1}{N} \frac{dN}{dt}$, is retained. In the rebalance portion of the flux-shape calculation, the time dependence of the flux shape is retained. This is possible because the scalar flux shape is stored for successive time steps. The equation programmed in SIMMER-II is taken from Eq. (L-21) with the flux shape time-dependence terms omitted. The equation also is divided by Δz_j , yielding

TABLE L-I

DEFINITIONS AND LOCATIONS OF CALCULATIVE VARIABLES
USED IN THE TRANSPORT EQUATIONS

<u>Storage Location Pointer Name</u>	<u>Length of Array</u>	<u>Mathematical Terms</u>
LA1	IM	$2\pi (r_{i+1/2} + r_{i-1/2}) = \frac{A_{i+1/2j} + A_{i-1/2j}}{\Delta z_j}$
LA2	IM	$2\pi (r_{i+1/2} - r_{i-1/2}) = \frac{A_{i+1/2j} - A_{i-1/2j}}{\Delta z_j}$
LA3	IM	$\pi(r_{i+1/2}^2 - r_{i-1/2}^2) = B_i$
LA4	IM+1	$2\pi r_{i-1/2} = \frac{A_{i-1/2j}}{\Delta z_j}$
LA5	IM	$\pi (r_{i+1/2}^2 - r_{i-1/2}^2)$
LB1	JM	$\frac{1}{z_{j+1/2} - z_{j-1/2}}$
LYH	JM	$z_{j+1/2} - z_{j-1/2}$
LCM	MM	μ_m
LCE	MM	η_m
LW	MM	w_m
LWM	MM	$\mu_m * w_m$
LWE	MM	$\eta_m * w_m$
LAL1	MM	$\alpha_{m+1/2}/w_m$
LAL2	MM	$\alpha_{m-1/2}/w_m$

$$\psi_{ijmg} = \frac{(AA)'_{ijm} \psi_{i-1/2jmg} + (BB)'_{im} \psi_{ij-1/2mg} + (CC)'_{ijm} \psi_{ijm-1/2g} + S_{c,ijg}}{(AA)'_{ijm} + (BB)'_{im} + (CC)'_{ijm} + (CT)_{ijg}}, \quad (L-38)$$

where

$$\begin{aligned} (AA)'_{ijm} &= \frac{(AA)_{ijm}}{\Delta z_j} = |\pm \mu_m| (A_{i+1/2j} + A_{i-1/2j}) \frac{1}{\Delta z_j} \\ &= |\pm \mu_m| 2\pi(r_{i+1/2} - r_{i-1/2}) \\ &= A(LCM)*A(LA1), \end{aligned}$$

$$\begin{aligned} (BB)'_{im} &= \frac{(BB)_{im}}{\Delta z_j} = \frac{2 |\pm \eta_m| B_i}{z_{j+1/2} - z_{j-1/2}} \\ &= \frac{2\pi |\pm \eta_m| (r_{i+1/2}^2 - r_{i-1/2}^2)}{z_{j+1/2} - z_{j-1/2}} \\ &= 2*A(LB1)*A(LCE)*A(LA3), \end{aligned}$$

$$\begin{aligned} (CC)'_{ijm} &= \frac{(CC)_{ijm}}{\Delta z_j} = \frac{1}{w_m} (\alpha_{m+1/2} + \alpha_{m-1/2}) (A_{i+1/2j} - A_{i-1/2j}) \frac{1}{\Delta z_j} \\ &= \frac{2\pi}{w_m} (r_{i+1/2} - r_{i-1/2}) (\alpha_{m+1/2} + \alpha_{m-1/2j}) \\ &= A(LA2)*(A(LAL1) + A(LAL2)), \end{aligned}$$

$$\begin{aligned} S_{c,ijg} &= \frac{v_{ij}}{\Delta z_j} \left(\sum_{g'=1}^g \Sigma_{s,ijg' \rightarrow g} \phi_{ijg'} + \frac{1}{7} \frac{x_{pg}}{k_o} \sum_{g'=1}^{IGM} (\nu_p \Sigma_f)_{ijg'} \phi_{ijg'} \right. \\ &\quad \left. + \frac{1}{N} \sum_{d=1}^{IGD} x_{dg} \lambda_d C_{ijgd} \right), \end{aligned}$$

= A(LA5)*(ingroup scatter + downscatter + fission + delayed neutron sources)(Note the presence of γ as explained in Chap. V.), and

$$(CT)_{ijg} = (\Sigma_{t,ijg} + \frac{1}{V_g} \frac{1}{N} \frac{dN}{dt}) \frac{V_{ij}}{\Delta z_j}$$

$$= A(LA5)*(total\ cross\ section + inverse\ period\ "loss") \ .$$

The volume element V_{ij} is given by

$$V_{ij} = \pi(r_{i+1/2}^2 - r_{i-1/2}^2) (z_{j+1/2} - z_{j-1/2}) = A(LA5)*A(LYH) \ .$$

B. Calculative Expressions Based on "Weighted-Diamond" Differencing

The equation programmed in SIMMER-II is taken from Eq. (L-26) with the flux-shape time-dependence terms omitted. The equation also is divided by Δz_j , yielding

$$\psi_{ijmg} = \frac{N_W}{D_W} \ , \tag{L-39}$$

where

$$N_W = (AW*ALPH + BETAP) \text{ and}$$

$$D_W = (AA)'_{ijm} + (BB)'_{ijm} + (CC)'_{ijm} + (CT)_{ijg} \\ + AW*((CT)_{ijg} + (CC)'_{ijm} + A(LA4)*A(LCM) + 1.2(BB)'_{ijm}) \ ,$$

where $(AA)'_{ijm}$, $(BB)'_{ijm}$, $(CC)'_{ijm}$, $(CT)_{ijg}$, and V_{ij} are defined as in the previous section. The remaining terms are defined as follows.

$$\begin{aligned}
\text{ALPH} &= |\pm\mu_m| \frac{A_{i\mp 1/2j}}{\Delta z_j} \psi_{i\mp 1/2jmg} + \frac{1}{\Delta z_j} |\pm\eta_m| B_i \psi_{ij-1/2mg} \\
&+ (\text{CC})'_{ijm} \psi_{ijm-1/2g} + (\text{CT})_{ijg} \\
&= |\pm\mu_m| 2\pi r_{i-1/2} \psi_{i-1/2jmg} + \frac{2\pi |\pm\mu_m| (r_{i+1/2}^2 - r_{i-1/2}^2)}{z_{j+1/2} - z_{j-1/2}} \psi_{ij-1/2mg} \\
&+ (r_{i-1/2}^2 - r_{i-1/2}^2) \frac{1}{\gamma} S'_{ijg} \\
&+ 2\pi (r_{i+1/2} - r_{i-1/2}) \frac{\alpha_{m+1/2} + \alpha_{m-1/2}}{w_m} \psi_{ijm-1/2g} \\
&= \text{A(LCM)} * \text{A(LA4)} * \psi_{i-1/2jmg} + 2 * \text{A(LB1)} * \text{A(LCE)} * \text{A(LA3)} * \psi_{ij-1/2mg} \\
&+ \text{A(LA5)} * (\text{ingroup scatter} + \text{downscatter} + \text{fission} \\
&+ \text{delayed neutron sources}) \\
&+ \text{A(LA2)} * (\text{A(LAL1)} + \text{A(LAL2)}) \psi_{ijm-1/2g} \cdot
\end{aligned}$$

$$\begin{aligned}
\text{BETAP} &= |\pm\mu_m| \frac{(A_{i+1/2j} + A_{i-1/2j})}{\Delta z_j} \psi_{i\mp 1/2jmg} + \frac{2}{\Delta z_j} |\pm\mu_m| B_i \psi_{ij\mp 1/2mg} \\
&+ \frac{V_{ij}}{\Delta z_j} \frac{1}{\gamma} S'_{ijg} + \frac{A_{i+1/2j} - A_{i-1/2j}}{w_m \Delta z_j} (\alpha_{m+1/2} + \alpha_{m-1/2}) \psi_{ijm-1/2g} \\
&+ |\pm\mu_m| 2\pi (r_{i+1/2} - r_{i-1/2}) \psi_{i-1/2jmg}
\end{aligned}$$

$$\begin{aligned}
& + \frac{2\pi |\pm\eta_m| (r_{i+1/2}^2 - r_{i-1/2}^2)}{z_{j+1/2} - z_{j-1/2}} \psi_{ij-1/2mg} \\
& + (r_{i+1/2}^2 - r_{i-1/2}^2) \frac{1}{\gamma} S'_{ijg} \\
& + 2\pi (r_{i+1/2} - r_{i-1/2}) \left(\frac{\alpha_{m+1/2} - \alpha_{m-1/2}}{w_m} \right) \psi_{ijm-1/2g} \\
& = A(LCM) * A(LA1) \psi_{i-1/2jmg} \\
& = 2 * A(LB1) * A(LCE) * (A(LA3) \psi_{ij-1/2mg} + A(LA5) * (...sources...)) \\
& + A(LA2) * (A(LAL1) + A('LAL2)) \psi_{ijm-1/2g} .
\end{aligned}$$

Let

$$\begin{aligned}
BETA &= 2 * BETAP - (AA)'_{ijm} + (BB)'_{ijm} \\
&+ (CC)'_{ijm} + (CT) * MAX(\psi_{i-1/2jmg}, \psi_{ij-1/2mg}) ;
\end{aligned}$$

then,

$$AW = BETA / ALPH \text{ if } BETA < 0 \text{ and}$$

$$AW = 0 \text{ if } BETA \geq 0 .$$

Note, if $AW = 0$, the flux is computed using the standard diamond formula; that is, Eq. (L-39) reduces to Eq. (L-38) if $AW = 0$.

C. Treatment of the Fine-Mesh Rebalance Equations

Equations (L-36) form a set of $I_{\max} * J_{\max}$ equations. In matrix form, the familiar five-diagonal band structure is obtained by numbering the unknowns, f_{ij} , sequentially in order of increasing i , then increasing j . The following matrix elements are identified by collecting terms in Eq. (L-37).

(1) Diagonal vector

$$\begin{aligned} & \left(\Sigma_{t,ijg} + \frac{1}{v_g} \frac{1}{N} \frac{dN}{dt} + \frac{1}{\Delta t} \frac{1}{v_g} \right) \phi_{ijg} v_{ij} + FR_{i+1/2j}^u \\ & + FL_{i-1/2j}^u + FU_{ij+1/2}^u + FD_{ij-1/2}^u \end{aligned}$$

(2) Upper super-diagonal

$$-FL_{i+1/2j}^u$$

(3) Lower super-diagonal

$$-FR_{i-1/2j}^u$$

(4) Upper off-diagonal

$$-FD_{ij+1/2}^u$$

(5) Lower off-diagonal

$$-FU_{ij-1/2}^u$$

(6) Source

$$\frac{1}{\gamma} (S'_{ijg} + \frac{1}{\Delta t} \frac{1}{v_g} \phi_{ijg}^{u,k-1}) v_{ij}$$

IV. S_n CONSTANTS AND ANGULAR ORIENTATION

For the neutron transport equation [Eq. (III-256)], the angular directions are divided into M intervals per quadrant. For a quadrature order of ISN, $M = (ISN)*(ISN + 2)/8$, positive pairs of direction cosines (μ_m, η_m) must be available. They are supplied by the user or, for ISN up to 16, they can be obtained from the subroutine SNCON. If provided by the user, the numbers should be supplied first for the largest η (one value of μ), second for the next largest η (two values of μ , the largest first), then for the third largest η (three values of μ , in order of decreasing magnitude), and so on. Only one quadrant of directions is used, and a 90° rotational symmetry is assumed to perform calculations for the other three quadrants.

The ordering of the direction cosines for ISN = 4 and ISN = 8 is shown in Figs. L-1 and L-2. The numbers in parentheses for ISN = 4 indicate the actual numbering 1,2,...,M, of the direction cosines. The other numbers indicate the order of progression through the angular mesh for both ISN = 4 and ISN = 8. In Fig. L-1, the directions are shown lying on levels of η , but this is not strictly necessary. However, it does reduce the ω derivative to a purely μ derivative. Almost the only restriction on the direction set is the number of directions. An arrangement of 1,1,2,6 directions, respectively, on each of the η levels could be used instead of the 1,2,3,4 sequence of directions shown, but ISN/2 η levels are required. A value of zero for η should be avoided. The quadrature weights w_m are checked to ensure that they sum to one-fourth (because only one quadrant of directions is entered). In Figs. L-1 and L-2, the directions in which the step function scheme is used in the angle extrapolation are marked with a cross.

REFERENCES

1. B. G. Carlson, and K. D. Lathrop, "Transport Theory - The Method of Discrete Ordinates," Chap. 3 in Computing Methods in Reactor Physics, H. Greenspan, C. N. Kelber, and D. Okrent, Eds. (Gordon and Breach, New York, 1968).
2. G. I. Bell and S. Glasstone, "Nuclear Reactor Theory," (New York, Van Nostrand Reinhold, 1970).

3. B. G. Carlson, "Transport Theory: Discrete Ordinates Quadrature Over the Unit Sphere," Los Alamos Scientific Laboratory report LA-4544 (1970).
4. B. G. Carlson, "Tables of Equal Weight Quadrature EQ_N Over the Unit Sphere," Los Alamos Scientific Laboratory report LA-4734 (1971).

APPENDIX M

CALCULATIVE APPROACH AND TREATMENT OF THE QUASI-STATIC AMPLITUDE EQUATION

The quasi-static approach used to solve the time-dependent neutron transport or diffusion equations is described in Chap. V. In the quasi-static approach, the neutron flux is separated into a space-, energy-, and time-dependent flux-shape equation coupled to a set of time-dependent amplitude equations. The amplitude equation set contains most of the time dependence of the flux. Although this equation set can be solved by any of the usual methods used for ordinary differential equations, the calculative effort may be significant because of the mathematical stiffness of the equation set. The method used by Kaganove¹ has proven to be effective and efficient on the closely related "point-kinetics" equations. Therefore, the Kaganove method has been adopted for SIMMER-II by generalization to account for additional time dependence in the amplitude-equation coefficients. The equations used in SIMMER-II are derived and the calculative approach is explained in this appendix.

I. INTERFACE WITH THE OVERALL QUASI-STATIC ALGORITHM

The amplitude equation must be solved at several points in the quasi-static algorithm. Extrapolation of the power in preparation for a series of fluid dynamics calculative steps requires solution of the amplitude equation based on extrapolated parameters. After the fluid-dynamics calculation, the amplitude-equation solution is recalculated based on an updated reactivity. Each gamma iteration at the end of each flux shape time step requires a series of amplitude equation solutions, each corresponding to a single reactivity step.

In each of these applications, the time dependence of the amplitude equation coefficients is determined previously and is represented in the form of (parabolic) polynomials. The time domain over which the solution is required is a single reactivity step, requiring many amplitude equation steps because of the "stiffness" of the equation set.

Input to the amplitude-equation algorithm consists of the reactivity time-step size; parabolic coefficients for ratios of the reactivity and the delayed-neutron fraction to the generation time; the initial power; and the initial delayed-neutron precursor concentrations. Output consists of the final power, the

delayed neutron precursor concentrations at the end of the reactivity step, and the coefficients used to integrate the spatial delayed neutron precursor concentrations.

II. DERIVATION OF CALCULATIVE EQUATIONS

The amplitude-equation set used in the quasi-static algorithm is given by

$$\frac{dN(t')}{dt'} = \frac{\rho(t') - \beta_T(t')}{\Lambda(t')} N(t') + \sum_{d=1}^{IGD} \lambda_d c_d(t') \quad \text{and} \quad (M-1)$$

$$\dot{c}_d(t') = \frac{\beta_d(t')}{\Lambda(t')} N(t') - \lambda_d c_d(t') \quad , \quad (M-2)$$

where $\beta_d(t')$ = effective delayed-neutron fractions for delayed group d,

$$\beta_T(t') = \sum_{d=1}^{IGD} \beta_d(t'),$$

$\Lambda(t')$ = effective prompt-neutron generation time,

$\rho(t')$ = reactivity,

$\lambda_d(t')$ = decay constant for delayed group d,

IGD = number of delayed groups,

$c_d(t')$ = effective delayed-neutron-precursor concentrations,

$N(t')$ = quasi-static amplitude function, and

t' = time relative to the beginning of the reactor transient.

Summation of Eq. (M-2) over delayed groups and substitution into Eq. (M-1) yields

$$\begin{aligned} \frac{dN(t')}{dt'} &= \frac{\rho(t') - \beta_T(t')}{\Lambda(t')} N(t') - \sum_{d=1}^{IGD} \left\{ \dot{c}_d(t') - \frac{\beta_d(t')}{\Lambda(t')} N(t') \right\} \\ &= \frac{\rho(t')}{\Lambda(t')} N(t') - \sum_{d=1}^{IGD} \dot{c}_d(t') \quad . \end{aligned} \quad (M-3)$$

Equations (M-2) and (M-3) are solved over the time domain of a reactor transient; thus, t' is a continuous variable and refers to the time since the beginning of the transient. In the SIMMER-II neutronics solution, time is discretized and subdivided into flux-shape time steps, which in turn are subdivided into reactivity steps. Thus, it is convenient to consider the solution of Eqs. (M-2) and (M-3) over a single reactivity step; that is,

$$t'_1 \leq t' \leq t'_N ,$$

where t'_1 and t'_N denote the end points of a reactivity step. For the purpose of numerically solving Eqs. (M-2) and (M-3) the reactivity step must be subdivided further into discrete intervals. Thus, the reactivity step is subdivided into $N-1$ amplitude steps where

$$t'_1 \leq t'_i < t'_{i+1} \leq t'_N .$$

The amplitude-equation set is solved separately for each amplitude step based on the use of analytic approximations to the solution of the differential equations. Thus, it is convenient to have a continuous variable representing time relative to the beginning of each amplitude step; therefore, the time variable t is defined as

$$t = t' - t'_i ,$$

where $t'_i \leq t' \leq t'_{i+1}$. Thus, integration of Eq. (M-3) over the amplitude step t'_i to t'_{i+1} yields

$$N(t) - N_0 = \int_0^t \frac{\rho}{\Lambda} N(\tau) d\tau - \sum_{d=1}^{IGD} (c_d(t) - c_{d0}) , \quad (M-4)$$

where $N_0 = N(t = 0) = N(t' = t'_i)$ and $c_{d0} = c_d(t = 0) = c_d(t' = t'_i)$ are the amplitude and the delayed-neutron precursor concentrations at the beginning of the

amplitude step, respectively. Integration of Eq. (M-2) by variation of parameters gives

$$c_d(t) - c_{d0} = \int_0^t \frac{\beta_d}{\Lambda} N(\tau) e^{-\lambda_d(t-\tau)} d\tau + c_{d0} (e^{-\lambda_d t} - 1) . \quad (M-5)$$

The following parabolic functions describe the time variation of the amplitude-equation set coefficients over the reactivity step.

$$\left(\frac{\rho}{\Lambda}\right)' = \rho'_0 + \rho'_1 t' + \rho'_2 t'^2 , \quad \text{and} \quad (M-6)$$

$$\left(\frac{\beta_d}{\Lambda}\right)' = \beta_{d0} + \beta_{d1} t' + \beta_{d2} t'^2 , \quad (M-7)$$

where $t'_1 \leq t' \leq t'_N$.

The following parabolic functions describe the time variation of the unknown amplitude solution and the known equation set coefficients over the amplitude step.

$$N = N_0 + N_1 t + N_2 t^2 , \quad (M-8)$$

$$\frac{\rho}{\Lambda} = \rho_0 + \rho_1 t + \rho_2 t^2 , \quad \text{and} \quad (M-9)$$

$$\frac{\beta_d}{\Lambda} = \beta_{d0} + \beta_{d1} t + \beta_{d2} t^2 , \quad (M-10)$$

where $t = t' - t'_i$ with $t'_i \leq t' \leq t'_{i+1}$, as above.

The two sets of coefficient functions are identical over the amplitude step; thus,

$$\rho_0 + \rho_1 t + \rho_2 t^2 = \rho'_0 + \rho'_1 (t + t'_i) + \rho'_2 (t + t'_i)^2 \quad , \quad \text{and}$$

$$\beta_{d0} + \beta_{d1} t + \beta_{d2} t^2 = \beta'_{d0} + \beta'_{d1} (t + t'_i) + \beta'_{d2} (t + t'_i)^2 \quad .$$

The relationship between the coefficients of the two sets of coefficient functions is determined by matching powers of t ; thus,

$$\rho_0 = \rho'_0 + \rho'_1 t'_i + \rho'_2 t'^2_i$$

$$\rho_1 = \rho'_1 + 2 \rho'_2 t'_i \quad , \quad \text{and}$$

$$\rho_2 = \rho'_2$$

and, similarly,

$$\beta_{d0} = \beta'_{d0} + \beta'_{d1} t'_i + \beta'_{d2} t'^2_i \quad ,$$

$$\beta_{d1} = \beta'_{d1} + 2 \beta'_{d2} t'_i \quad , \quad \text{and}$$

$$\beta_{d2} = \beta'_{d2} \quad .$$

Substitution of Eqs. (M-5), (M-8), (M-9), and (M-10) into Eq. (M-4) yields

$$\begin{aligned} N_1 t + N_2 t^2 &= \int_0^t (\rho_0 + \rho_1 \tau + \rho_2 \tau^2) (N_0 + N_1 \tau + N_2 \tau^2) d\tau \\ &- \sum_{d=1}^{\text{IGD}} \int_0^t (\beta_{d0} + \beta_{d1} \tau + \beta_{d2} \tau^2) (N_0 + N_1 \tau + N_2 \tau^2) e^{-\lambda_d(t-\tau)} d\tau \\ &+ \sum_{d=1}^{\text{IGD}} c_{d0} (1 - e^{-\lambda_d t}) \quad , \end{aligned}$$

$$\begin{aligned}
&= N_0 \left(\rho_0 t + \rho_1 \frac{t^2}{2} + \rho_2 \frac{t^3}{3} \right) + N_1 \left(\rho_0 \frac{t^2}{2} + \rho_1 \frac{t^3}{3} + \rho_2 \frac{t^4}{4} \right) \\
&+ N_2 \left(\rho_0 \frac{t^3}{3} + \rho_1 \frac{t^4}{4} + \rho_2 \frac{t^5}{5} \right) \\
&- N_0 \sum_{d=1}^{\text{IGD}} \int_0^t (\beta_{d0} + \beta_{d1} \tau + \beta_{d2} \tau^2) e^{-\lambda_d(t-\tau)} d\tau \\
&- N_1 \sum_{d=1}^{\text{IGD}} \int_0^t (\beta_{d0} \tau + \beta_{d1} \tau^2 + \beta_{d2} \tau^3) e^{-\lambda_d(t-\tau)} d\tau \\
&- N_2 \sum_{d=1}^{\text{IGD}} \int_0^t (\beta_{d0} \tau^2 + \beta_{d1} \tau^3 + \beta_{d2} \tau^4) e^{-\lambda_d(t-\tau)} d\tau \\
&+ \sum_{d=1}^{\text{IGD}} c_{d0} (1 - e^{-\lambda_d t}) .
\end{aligned}$$

Next, the $I_{dm}(t)$ functions are defined implicitly by

$$t^{m+1} I_{dm}(t) = \int_0^t e^{-\lambda_d(t-\tau)} \tau^m d\tau \quad . \quad (\text{M-11})$$

To limit computational loss of significant figures, the "I" functions are evaluated by two different approaches, depending on the size of $\lambda_d t$. If $\lambda_d t \geq 1$, the recurrence relation,

$$\lambda_d t I_{dm+1}(t) = 1 - (m+1) I_{dm} \quad ,$$

is used. If $\lambda_d t < 1$ the highest order "I" used, I_{d4} , is calculated first, by means of the series expansion

$$I_{dm} = \frac{1}{m+1} - \frac{\lambda_d t}{(m+2)(m+1)} + \frac{(\lambda_d t)^2}{(m+3)(m+2)(m+1)} - \dots$$

The remaining functions then are determined by the recursion relation,

$$m I_{dm-1} = 1 - \lambda_d t I_{dm} \quad .$$

By substitution of the appropriate "I" functions, Eq. (M-4) is transformed further into,

$$\begin{aligned} & N_1 \left[t - \left(\rho_0 \frac{t^2}{2} + \rho_1 \frac{t^3}{3} + \rho_2 \frac{t^4}{4} \right) \right. \\ & \quad + \sum_{d=1}^{IGD} \left(\beta_{d0} t^2 I_{d1} + \beta_{d1} t^3 I_{d2} + \beta_{d2} t^4 I_{d3} \right) \Big] \\ & + N_2 \left[t^2 - \left(\rho_0 \frac{t^3}{3} + \rho_1 \frac{t^4}{4} + \rho_2 \frac{t^5}{5} \right) \right. \\ & \quad + \sum_{d=1}^{IGD} \left(\beta_{d0} t^3 I_{d2} + \beta_{d1} t^4 I_{d3} + \beta_{d2} t^5 I_{d4} \right) \Big] \\ & = N_0 \left[\left(\rho_0 t + \rho_1 \frac{t^2}{2} + \rho_2 \frac{t^3}{3} \right) \right. \\ & \quad - \sum_{d=1}^{IGD} \left(\beta_{d0} t I_{d0} + \beta_{d1} t^2 I_{d1} + \beta_{d2} t^3 I_{d2} \right) \Big] \\ & \quad + \sum_{d=1}^{IGD} \lambda_d c_{d0} t I_{d0} \quad . \end{aligned} \tag{M-12}$$

If Eq. (M-12) is evaluated at $t = \Delta t$ and $t = \Delta t/2$, two equations for two unknowns result. Further, if Eq. (M-12) is evaluated at $t = \Delta t/4$, the unknowns N_1 and N_2 can be solved using the $t = \Delta t/2$ and $t = \Delta t/4$ evaluations. Thus, two independent solutions can be compared and used as a basis for error and time-step control.

The delayed-neutron precursor concentrations, Eqs. (M-5), are solved similarly. Substituting Eqs. (M-8) and (M-10) into Eq. (M-5) yields

$$\begin{aligned}
 c_d(t) - c_{d0} &= \int_0^t (\beta_{d0} + \beta_{d1} \tau + \beta_{d2} \tau^2) (N_0 + N_1 \tau + N_2 \tau^2) e^{-\lambda_d(t-\tau)} d\tau \\
 &+ c_{d0}(e^{-\lambda_d t} - 1) \\
 &= N_0 (\beta_{d0} t I_{d0} + \beta_{d1} t^2 I_{d1} + \beta_{d2} t^3 I_{d2}) \\
 &+ N_1 (\beta_{d0} t^2 I_{d1} + \beta_{d1} t^3 I_{d2} + \beta_{d2} t^4 I_{d3}) \\
 &+ N_2 (\beta_{d0} t^3 I_{d2} + \beta_{d1} t^4 I_{d3} + \beta_{d2} t^5 I_{d4}) \\
 &+ \lambda_d c_{d0} t I_{d0} \quad .
 \end{aligned} \tag{M-13}$$

Equation (M-13) is evaluated readily when N_1 and N_2 have been determined.

III. SOLUTION ALGORITHM

The algorithm used to solve the amplitude equation set is divided into three main segments. The calculative flow is illustrated in Fig. M-1. The approach is based on precalculation of the basic time-step and integration data needed to solve the amplitude equation for different time-step sizes. This allows efficient solution of the amplitude equation for different trial time steps. The main calculative segments are as follows.

1. The time-step ($\Delta t = t'_{i+1} - t'_i$) matrix is determined for eight time step sizes. The largest time step is 1/8 of the reactivity step; the smallest is 1/256 of the reactivity step. Powers of the time steps are also precalculated; the first through the fifth powers are needed to evaluate Eqs. (M-12) and (M-13). Next the "I" integral functions, defined by Eq. (M-11), are evaluated; a two-dimensional matrix is generated for each row (different powers of the same time-step size) in the Δt matrix, with $\Delta t^{m+1} I_{dm}$ evaluated for $0 \leq m \leq 4$ and $1 \leq d \leq \text{IGD}$. Two methods are used depending on the size of $\lambda_d \Delta t$. For $\lambda_d \Delta t \geq 1$, the

Fig. M-1.

Calculative flow for the amplitude equation solution.

first term, $\Delta t I_{d0}$, is calculated directly, and a recurrence relation is used to generate the other $\Delta t^{m+1} I_{dm}$. For $\lambda_d \Delta t < 1$, the last term, $\Delta t^5 I_{d4}$, is computed from a series, and a recursion relation is used to generate the remaining $\Delta t^{m+1} I_{dm}$.

2. Parabolic functions for ρ/Λ and β_i/Λ are calculated from tabulated data.
3. Three evaluations of Eq. (M-12) for a trial time step and 1/2 and 1/4 of the trial time step are performed. Two separate solutions are formed from three evaluations using Cramer's rule to solve each system of two equations in two unknowns. Truncation error is estimated by the difference in these two solutions. If the error is excessive, the trial time step is reduced (based on the next smaller precomputed values), and this step is repeated. The iteration is terminated when the upper error limit is satisfied. The time step is increased if the error falls below a lower limit. After the iteration is converged for each amplitude-equation time step, Eq. (M-13) is used to evaluate the delayed-neutron precursor concentrations. The series of amplitude equation time steps then is continued until the solution procedure reaches the final time (that is, the end of the reactivity step).

IV. INTEGRATION OF SPATIAL DELAYED NEUTRON PRECURSOR EQUATIONS

The flux factorization that is the basis for the quasi-static approach produces two equation sets when applied to the time-dependent neutron-diffusion or transport equation. The amplitude-equation set (which is the subject of this appendix) and the flux-shape equation set (the solution of which is the subject of Appendices I and L) both involve related terms. The flux-shape equation set includes a spatial delayed-neutron source term in the flux-shape equation and a spatial delayed-neutron precursor balance equation. The integral of the spatial delayed-neutron precursor is formally equal to the effective delayed-neutron precursor concentration used in the amplitude-equation set; that is,

$$\int_0^\infty \int_R \phi^*(\vec{r}, E) \lambda_d(E) C_d(\vec{r}, t) d\vec{r} dE = c_d(t) \quad . \quad (M-14)$$

where ϕ^* is the initial adjoint scalar flux, (used as a weighting function for all inner products involved in coupling the shape and amplitude equation sets) and $\vec{R}$ is the region of solution of the spatial-equation set. $\chi_d(E)$ is the delayed neutron emission spectrum associated with delayed family d, and $C_d(\vec{r}, t)$ is the spatial delayed neutron precursor concentration.

The balance equation for the spatial precursor concentration is given by

$$\frac{\partial C_d(\vec{r}, t)}{\partial t} = N(t) F_d^d(\vec{r}, t) - \lambda_d C_d(\vec{r}, t) \quad , \quad (M-15)$$

where $F_d^d(\vec{r}, t)$ is the delayed neutron precursor source shape term,

$$F_d^d(\vec{r}, t) = \frac{1}{k_0} \sum_{g=1}^{IGM} \nu_d \Sigma_{f,g}(\vec{r}, t) \phi_g(\vec{r}, t) \quad , \quad (M-16)$$

with terms as defined in Chap. V. Each term in Eq. (M-15) has a corresponding term in Eq. (M-2). Equation (M-15) is solved differently than Eq. (M-2), because the flux-shape equation set is solved with much larger time steps than the amplitude equation. However, Eq. (M-15) can be solved using integrating factors determined during the solution of Eqs. (M-1) and (M-2). This makes a more accurate solution of Eq. (M-15) than would otherwise be possible with the large time steps. Thus, integration of Eq. (M-15) in analogy with the integration of Eq. (M-2) to yield (M-5) results in

$$C_d(\vec{r}, t) = \int_0^t N(\tau) F_d^d(\vec{r}, \tau) e^{-\lambda_d(t-\tau)} d\tau + C_d(\vec{r}, 0) e^{-\lambda_d t} \quad . \quad (M-17)$$

The delayed-neutron source is calculated only at reactivity steps with an assumed linear time dependence at intermediate times; that is,

$$F_d^d(\vec{r}, \tau') = F_d^d(\vec{r}, t'_1) \left(1 - \frac{\tau' - t'_1}{t'_N - t'_1}\right) + F_d^d(\vec{r}, t'_N) \frac{\tau' - t'_1}{t'_N - t'_1} \quad , \quad (M-18)$$

where τ is relative to the beginning of the transient, similar to t' .

Substituting Eq. (M-18) into (M-17) and assuming that the time integration is over the amplitude step, t'_i to t'_{i+1} , yields

$$C_d(\vec{r}, t'_{i+1}) - C_d(\vec{r}, t'_i) = \int_{t'_i}^{t'_{i+1}} N(\tau' - t'_i) F_d^d(\vec{r}, \tau') e^{-\lambda_d(t'_{i+1} - \tau')} d\tau' \\ + C_d(\vec{r}, t'_i)(e^{-\lambda_d(t'_{i+1} - t'_i)} - 1)$$

Substituting Eq. (M-8) for $N(t)$ and Eq. (M-18) for $F_d^d(\vec{r}, \tau)$ leads to

$$C_d(\vec{r}, t'_{i+1}) - C_d(\vec{r}, t'_i) = F_d^d(\vec{r}, t'_i) \int_{t'_i}^{t'_{i+1}} \{N_0 + N_1(\tau' - t'_i) \\ + N_2(\tau' - t'_i)^2\} e^{-\lambda_d(t'_{i+1} - \tau')} d\tau' \\ - F_d^d(\vec{r}, t'_i) \int_{t'_i}^{t'_{i+1}} (N_0 + N_1(\tau' - t'_i) \\ + N_2(\tau' - t'_i)^2) \left(\frac{\tau' - t'_i}{t'_N - t'_i} \right) e^{-\lambda_d(t'_{i+1} - \tau')} d\tau' \\ + F_d^d(\vec{r}, t'_N) \int_{t'_i}^{t'_{i+1}} \{N_0 + N_1(\tau' - t'_i) \\ + N_2(\tau' - t'_i)^2\} \left(\frac{\tau' - t'_i}{t'_N - t'_i} \right) e^{-\lambda_d(t'_{i+1} - \tau')} d\tau' \\ + C_d(\vec{r}, t'_i)(e^{-\lambda_d(t'_{i+1} - t'_i)} - 1) \quad (M-19)$$

Using previous definitions, the fact that $t'_{i+1} - t'_i = t_{i+1} - t_i \equiv \Delta t$, and making the transformation, $\tau = \tau' - t'_i$, the right-hand side of Eq. (M-19) becomes the following.

$$\text{First term} = F_d^d(\vec{r}, t'_i) \int_0^{\Delta t} (N_0 + N_1 \tau + N_2 \tau^2) e^{-\lambda_d(t'_{i+1} - \tau - t'_i)} d\tau \\ = F_d^d(\vec{r}, t'_i) [N_0 \Delta t_i I_{d0} + N_1 \Delta t_i^2 I_{d1} + N_2 \Delta t_i^3 I_{d2}]$$

$$\begin{aligned}
\text{Second term} &= F_d^d(\vec{r}, t'_1) \int_0^{\Delta t'_1} (N_0 + N_1 \\
&\quad + N_2 r^2) \left(\frac{r + t'_i - t'_1}{t'_N - t'_1} \right) e^{-\lambda_d(t'_{i+1} - r - t'_i)} dr \\
&= F_d^d(\vec{r}, t'_1) \left[\frac{1}{t'_N - t'_1} (N_0 \Delta t_i^2 I_{d1} + N_1 \Delta t_i^3 I_{d2} + N_2 \Delta t_i^4 I_{d3}) \right. \\
&\quad \left. + \left(\frac{t'_i - t'_1}{t'_N - t'_1} \right) (N_0 \Delta t_i I_{d0} + N_1 \Delta t_i^2 I_{d1} + N_2 \Delta t_i^3 I_{d2}) \right] .
\end{aligned}$$

$$\begin{aligned}
\text{Third term} &= F_d^d(\vec{r}, t'_N) \int_0^{\Delta t_i} (N_0 + N_1 \\
&\quad + N_2 r^2) \left(\frac{r + t'_i - t'_1}{t'_N - t'_1} \right) e^{-\lambda_d(t'_{i+1} - r - t'_i)} dr \\
&= F_d^d(\vec{r}, t'_N) \left[\frac{1}{t'_N - t'_1} (N_0 \Delta t_i^2 I_{d1} + N_1 \Delta t_i^3 I_{d2} + N_2 \Delta t_i^4 I_{d3}) \right. \\
&\quad \left. + \left(\frac{t'_i - t'_1}{t'_N - t'_1} \right) (N_0 \Delta t_i I_{d0} + N_1 \Delta t_i^2 I_{d1} + N_2 \Delta t_i^3 I_{d2}) \right] .
\end{aligned}$$

Combining terms leads to

$$\begin{aligned}
C_d(\vec{r}, t'_{i+1}) &= F_d^d(\vec{r}, t'_1) \left[N_0 \left(\frac{t'_N - t'_i}{t'_N - t'_1} \Delta t_i I_{d0} - \frac{1}{t'_N - t'_1} \Delta t_i^2 I_{d1} \right) \right. \\
&\quad + N_1 \left(\frac{t'_N - t'_i}{t'_N - t'_1} \Delta t_i^2 I_{d1} - \frac{1}{t'_N - t'_1} \Delta t_i^3 I_{d2} \right) \\
&\quad + N_2 \left(\frac{t'_N - t'_i}{t'_N - t'_1} \Delta t_i^3 I_{d2} - \frac{1}{t'_N - t'_1} \Delta t_i^4 I_{d3} \right) \Big] \\
&\quad + F_d^d(\vec{r}, t'_N) \left[N_0 \left(\frac{t'_i - t'_1}{t'_N - t'_1} \Delta t_i I_{d0} + \frac{1}{t'_N - t'_1} \Delta t_i^2 I_{d1} \right) \right.
\end{aligned}$$

$$\begin{aligned}
& + N_1 \left(\frac{t'_i - t'_1}{t'_N - t'_1} \Delta t_i^2 I_{d1} - \frac{1}{t'_N - t'_1} \Delta t_i^3 I_{d2} \right) \\
& + N_2 \left(\frac{t'_i - t'_1}{t'_N - t'_1} \Delta t_i^3 I_{d2} - \frac{1}{t'_N - t'_1} \Delta t_i^4 I_{d3} \right) \\
& + C_d(\vec{r}, t'_i) e^{-\lambda_d \Delta t_i} .
\end{aligned} \tag{M-20}$$

Solving for the spatial precursor shape at the end of the reactivity step, $C_d(\vec{r}, t'_N)$, is performed by solving Eq. (M-20) for each successive amplitude step in the reactivity step; that is,

$$\begin{aligned}
C_d(\vec{r}, t'_N) &= F_d^d(\vec{r}, t'_1) ([...]_N + [...]_{N-1} e^{-\lambda_d \Delta t_{N-1}} + [...]_{N-2} e^{-\lambda_d (\Delta t_{N-1} + \Delta t_{N-2})} + \dots) \\
&+ F_d^d(\vec{r}, t'_N) ([...]_N + [...]_{N-1} e^{-\lambda_d \Delta t_{N-1}} + [...]_{N-2} e^{-\lambda_d (\Delta t_{N-1} + \Delta t_{N-2})} + \dots) \\
&+ C_d(\vec{r}, t'_1) e^{-\lambda_d (t'_N - t'_1)} ,
\end{aligned}$$

where the bracketed quantities are those from Eq. (M-20) with subscripts added to indicate the time step for which they are to be evaluated. Thus, formally, $C_d(\vec{r}, t'_N)$ becomes

$$\begin{aligned}
C_d(\vec{r}, t'_N) &= F_d^d(\vec{r}, t'_1) \sum_{i=1}^N [N_0 ((t'_N - t'_i) \Delta t_i I_{d0} - \Delta t_i^2 I_{d1}) \\
&+ N_1 ((t'_N - t'_i) \Delta t_i^2 I_{d1} - \Delta t_i^3 I_{d2}) \\
&+ N_2 ((t'_N - t'_i) \Delta t_i^3 I_{d2} - \Delta t_i^4 I_{d3})] \frac{e^{-\lambda_d (t'_N - t'_i)}}{t'_N - t'_i} \\
&+ F_d^d(\vec{r}, t'_N) \sum_{i=1}^N [N_0 ((t'_i - t'_1) \Delta t_i I_{d0} - \Delta t_i^2 I_{d1}) \\
&+ N_1 ((t'_i - t'_1) \Delta t_i^2 I_{d1} - \Delta t_i^3 I_{d2})
\end{aligned}$$

$$\begin{aligned}
& + N_2 \{ (t'_i - t'_1) \Delta t_i^3 I_{d2} - \Delta t_i^4 I_{d3} \} \frac{e^{-\lambda_d(t'_N - t'_i)}}{t'_N - t'_1} \\
& + C_d(\vec{r}, t'_1) e^{-\lambda_d(t'_N - t'_1)} .
\end{aligned}$$

The spatial integral of the spatial delayed neutron precursor concentrations, $C_d(\vec{r}, t)$, are formally identical to the effective delayed neutron precursor concentrations, $c_d(t)$, for each family, d . However, in practice, numerical errors introduced by approximations such as the assumption of a linear delayed fission source between reactivity time steps, round-off errors, and the effect of different calculative algorithms lead to differences. Because of the integral nature of the time dependence of the delayed neutron precursors, the effect of such errors is cumulative.

To maintain the correspondence between the solutions for $C_d(\vec{r}, t)$ and $c_d(t)$ the functions are rebalanced periodically; failure to do this leads to a nonconvergence of the "gamma" iteration at the end of the flux-shape time step, especially in subcritical situations where the delayed neutron source is a significant portion of the total source. Because the algorithms treating the effective delayed neutron precursor concentrations are considered to be the most accurate, the spatial delayed neutron precursor concentrations are scaled so that their spatial integrals agree with the effective quantities. This operation is performed at each "gamma" iteration at the end of the flux-shape time steps.

REFERENCE

1. J. J. Kaganove, "Numerical Solution of the One-Group, Space-Independent Reactor Kinetics Equations for Neutron Density Given the Excess Reactivity," Argonne National Laboratory Report ANL-6132 (February 1960).

APPENDIX N

SIMMER-II SUBROUTINES, FUNCTIONS, AND COMMON DECKS

Table N-I lists all the subroutines and functions in the SIMMER-II source file. The routines are listed in the order of their appearance. Table N-II lists all the common decks in the SIMMER-II source file in the order of their appearance. Table N-III lists the overlay entry points and their functions. All CLAMS library routines are from Ref. 1.

TABLE N-I
SUBROUTINE FUNCTIONS

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
SIMMER	Main driver of the SIMMER-II program package. Establishes maximum problem size, assigns logical units, and calls the input and calculation driver overlays.
TSDRIV	Transient calculation driver. Alternatively calls the transient neutronics calculation overlay and the transient fluid-dynamics calculation overlay.
TIMSTP	Performs the time-step selection for the SIMMER-II space-dependent neutronics.
NPUT1	Calls the appropriate input routines.
NPUT11	Reads and edits the basic neutronics parameter data and calculates storage locations in the container array and in bulk storage.
NPUT12	Calls INXS to input and process cross sections.
INXS	Reads, edits, and selectively stores cross-section input from the standard interface file ISOTXS.
NPUT13	Calls INFF to input and process self-shielding factors.
INFF	Reads, edits, and selectively stores shielding factors from the standard interface file BRKOXS.
NPUT14	Calls IFINSN or SNCON to obtain S_n constants and calculates angular and alpha coefficients.

TABLE N-I (CONT)

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
SNCON	Internal library of S_n constants, S_2 through S_{16} .
IFINSN	Reads S_n constants from standard file SNCON.
NPUT15	Reads neutronics array data specified by input parameters, including delayed neutron data.
NPUT17	Reads header, title cards, and problem dimensions and assigns logical units.
NPUT1A	Initializes all constants used in SIMMER-II.
STOEDT	Checks for sufficient small- and large-core storage and prints table of core storage used.
GRIND	Controls the neutronics calculation. Calls SHUFFL, GRIND1, GRIND2, PKDRIV, and RITE.
GRIND1	Controls the calculation of the pointwise cross sections. Calls SHLDXS.
SHLDXS	Main routine for cross-section calculation. Calculates isotopic shielding factors, then mixes cell-wise macroscopic cross sections. Calls BINT4 if NEUPREP is on.
GRIND2	Controls the calculation of the flux shape. Calls OUTER.
OUTER	Performs fission source iteration and calculates source to each group. Calls OUTACC, CSGEN, and INNER.
OUTACC	Normalizes the fission source, checks for the fission source convergence, and prints the outer iteration summary. Performs the Chebyshev acceleration. Calls TRINIT.
TRINIT	Determines the data adjustment factors for the transient neutronics initialization.
REBAL	Calculates rebalance factors for discrete ordinates neutron transport solution. Not present if DIFF is on.
INNET	For the discrete ordinates method, calculates the total source for a group, performs space-angle mesh sweeps, and applies rebalance factors calculated by REBAL. Calls REBAL. Not present if DIFF is on.
FSOL	For the diffusion method, solves matrix of finite-difference equations for the flux solution. Not present if DIFF is off.

TABLE N-I (CONT)

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
INNED	For the diffusion method, reads the finite-difference coefficients and calls FSOL to determine the flux. Not present if DIFF is off.
CSGEN	Determines which cross sections are needed in either OUTER or INPROD, retrieves them from random access disk, and places them in LCM. Not present if COREXS is on.
PCALC	Updates the transient delayed-neutron precursor distribution.
FIXUP	For the transport option, corrects negative fluxes. Not present if FIXUP is off.
POWCAL	Calculates the power distribution. For the initial flux calculation, normalizes the fluxes and the delayed-neutron precursor distributions.
INPROD	Calculates the inner products necessary to determine amplitude equation coefficients. Calculates, edits, and stores pointwise reactivity contributors if RHOMAP is on.
TSPK	Calculates the solution to the amplitude equation.
FITZ	Fits a parabola to three points in time.
PKDRIV	Drives the calculation flow for the quasi-static method. Calls INPROD to determine amplitude equation coefficients, calls TSPK to determine the solution to the amplitude equation, calls POWCAL to determine component power densities, and updates arrays to calculate the new flux shape.
PCINT	Integrates the spatial precursor densities and renormalizes to agree with precursors from the amplitude equation.
EXTRAP	Performs linear extrapolation or interpolation in time.
SETUP	Reads fluid-dynamics input data.
SETOUT	Edits fluid-dynamics input data.
SETIN	Initializes certain fluid-dynamic variables.
REGINP	Reads and edits mesh-cell-dependent fluid-dynamics data.
EOSINP	Inverts the EOS to calculate the material thermodynamic state for initial conditions.

TABLE N-I (CONT)

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
SSPK	Reads power and reactivity worth tables and sets up initial point kinetics data used in the URANUS option.
HYDRO	Main fluid-dynamics driver program. Computes the fluid-dynamics time-step size. Calls the subroutines XCHANJ, EXFLUD, and IMFLUD. Performs a partial output edit and controls the problem finish time.
EDI LPT	Calls the subroutines FLMPAR, FL1DPT, FL2DPT, FL3DPT to plot output.
FLMPAR	Plots velocity vectors for each cell.
FL1DPT	Creates plots for one-dimensional calculations.
FL2DPT	Creates contour plots for two-dimensional calculations.
FL3DPT	Creates three-dimensional perspective plots for two-dimensional calculations.
FLWPF	Creates an array of a variable for writing out to the postprocessing file.
BUFSTF	Packs postprocessing array elements and fills up a larger buffer for the postprocessing file. Data are packed two to a word when the IBM option is off. If the IBM option is on, the "DBL" option is assumed to be on, and each double-precision data word is truncated to a single-precision word.
EDFLPR	Performs fluid-dynamics edits.
DBFLPR	Performs fluid-dynamics edit of temporary AASC variables calculated in EXFLUD.
EXFLUD	Driver routine for the explicit portion of the fluid-dynamics calculations. Calls EXFL1 and EXFL2.
EXFL1	Updates the structure components, the liquid-field internal energies, and the vapor densities in single-phase cells. Estimates end-of-cycle cell pressures and tests for can-wall failure.
EXFL2	Calculates coefficients for all terms required to solve Eqs. (IV-177) to (IV-180).
CANYLD	Calculates the yield strength in the can-wall failure model by linear interpolation.

IMFLUD	Performs the implicit portion of the fluid-dynamics calculations.
MATSOL	Solves a one-dimensional geometry matrix equation by the Crout-Choleskey method. When the MATSOL option is on, it solves a two-dimensional geometry matrix using the Successive Line Over-Relaxation (SLOR) method. When the MATSOL option is off, it solves a two-dimensional geometry matrix by the Approximate Inversion Method (AIM).
YCHANJ	Contains the structure-configuration model and acts as a driver subroutine for the mass, momentum, and energy-transport exchange-function calculations.
FAM	Obtains the momentum-exchange functions and the melting/freezing rates.
PHASE	Determines the liquid/vapor phase-transition rates for mesh cells with two-phase flow.
TSHTR	Calculates the structure- and liquid-field heat transfer and determines the liquid particle radii.
	GAUSS Solves the matrix equation $AX = B$ by the Gauss elimination method with complete pivoting. A is an M by M matrix; X and B are 1 by M matrices.
EXCHNG	Performs the multicomponent phase-transition calculation.
P4TI	Processes the input EOS tables for the tabular EOS.
E4	Calculates the initial EOS energies for the structure-, liquid-, and vapor-field energy components when the component temperatures are provided.
P4	Obtains pressures and temperatures from the energies and densities for the tabular EOS.
GE4	Provides the vapor energy, given the temperature and density for the tabular EOS.
SMGETI	Loads the SIMMER-II EOS tables from the SESAME tables.
LA401A	Determines quantities on the vaporization/sublimation curve for the tabular EOS.
LA302B	Determines cold curve data for the tabular equation of state.

TABLE N-I (CONT)

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
RATFN2	Interpolates a function of two variables and obtains the values of the function and its derivatives.
TBFCH	Picks appropriate tables from the SESAME table.
ISRCH	Finds an index in a table (tabular EOS routine).
INV301	Inverts a data string.
RATFN1	Interpolates a function of one variable and obtains the value of the function and its derivatives.
DRHODT	Calculates EOS derivatives for a vapor mixture and other quantities for the vaporization-condensation model.
P4ENER	Calculates the EOS for a vapor mixture and certain quantities along the saturation curve.
LA302S	Search/interpolate for pressure and temperature as functions of density and internal energy.
P4	Calculates EOS pressures and saturation conditions from the analytic EOS.
HTSPK	Performs the point kinetics solution if the URANUS option is used.
PREA	Calculates programmed reactivity or power used in URANUS option.
FEEDBK	Calculates reactivity feedback if the URANUS option is used.
HENERG	Calculates the component internal energy generation rates for the URANUS option.
DENST	Calculates material total macroscopic densities and temperatures for the space-dependent neutronics cross-section calculation.
ENERGY	Obtains the material-dependent internal energy generation rates from LCM and places them in the fluid-dynamics array.
INTIRP	A linear interpolation subroutine for interpolating the input power-vs-time or reactivity-vs-time tables in the URANUS option.

TABLE N-1 (CONT)

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
BINT4	CLAMS library subroutine to compute a cubic B-spline representation of pointwise data. Used by SHLDXS for optional B-spline interpolation of shielding factor temperature dependence.
BVALU	CLAMS library subroutine to evaluate the B-spline representation to obtain a given function value or any of its derivatives. Used by SHLDXS for optional B-spline interpolation of shielding factor temperature dependence.
BNFAC	CLAMS library subroutine to perform a LU factorization of a banded matrix.
BNSLV	CLAMS library subroutine to calculate the solution of a linear system given the LU factorization.
BSPVD	CLAMS library subroutine to evaluate B-spline basis functions. Called by BINT4.
BSPVN	CLAMS library subroutine to evaluate all nonzero B-spline basis functions. Called by BSPVD.
INTRV	CLAMS library subroutine to locate the proper tabulation points to interpolate from B-splines at a requested point. Called by BVALU.
R1MACH	CLAMS library subroutine to set up machine dependent constants for use by CLAMS subroutines.
SHUFFL	Swaps the SIMCOM common block and LCM for the neutronics and fluid-dynamics section of the code using random disk files.
ERROR	Prints error messages.
CLEAR	Sets a real array to a specified constant.
ICLEAR	Sets an integer array to a specified constant.
LCLEAR	Sets a literal array to a specified constant.
ECHECK	Compares the elements of a vector for equality.
REED	Performs binary read operations from disk or large-core memory.
ECRD	Simulates floating point data transfer between LCM and SCM for machines with only one memory.

TABLE N-I (CONT)

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
IECRD	Simulates integer transfer between LCM and SCM for machines with only one memory.
GETQ	Fetches system parameters. Used in SIMMER-II to get job time limit.
ABORT	Aborts a job and retains a copy of memory on disk.
DATE1	Gets the date from the system in A8 format.
CLOCK1	Obtains the wall clock time for the VAX/VMS system.
DATE	Obtains the date for the VAX/VMS system.
LOCF	Obtains the address of the variable for the VAX/VMS system.
SECOND	Obtains the elapsed CPU time for the VAX/VMS system.
CUMCPUSEC	Obtains the total CPU time from the VAX/VMS system.
SHIFT	A shift function for the VAX computer.
DEFILE	Opens a random file on the VAX computer.
RITE	Performs binary write operations from SCM to LCM or disk or from LCM to disk.
RTEDIT	Edits the execution times of certain routines.
TICKEK	Timing routine for the LTSS FTN compiler.
GRPHFTN	Initializes film writing routines for LTSS-FTN.
GSET	Manipulates external storage for graphics on LTSS-FTN.
KSADDR	Calculates the number of words between two variables inclusively.
MEMEXP(IA,N,I)	This subroutine dynamically expands the amount of memory assigned to the job by N single words. I is returned and indicates the first word of the allocated storage space when referenced to the arbitrary single word array IA; that is, IA(I), ..., IA(I+N-1) is the allocated storage space. If I = 0 is returned, the memory expansion failed. If the memory expansion is required for N double words referenced to a double word array A, the routine must be called by CALL MEMEXP (A, 2*N, I), and the returned index I (assumed to indicate a double word boundary) must be converted into $I' = (I+1)/2$; then A(I'), ..., A(I'+N-1) is the allocated space.

TABLE N-I (CONT)

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
LABS2R	Produces an eight-character code giving the number of years past leap year, the day of the year, and the time of day. The code labels all dumps produced from a given run and its restarts. This code appears at the beginning of the output file.
DMPCTL	Decides whether to read or write a restart file. A control program used only with overlays.
DMPRD	Reads from a restart file all the data required to restart SIMMER-II.
DMPWR	Writes to restart dump all the data required to restart SIMMER-II.
RDPLCM	Using BUFFER IN instructions, reads data from disk to LCM.
WDPLCM	Using BUFFER OUT instructions, writes data from LCM to disk.
HYDSET	Resets, by means of NAMELIST, certain fluid-dynamics quantities to be altered in subsequent restarts.
NEUSET	Resets, by means of NAMELIST, certain neutronics quantities to be altered in subsequent restarts.
IRANRE	Reads/writes from random disk locations into/from large-core locations.
IDONEQ	Tests completion of random input/output operations on disk.
FORQTS	Sets up a file request table and creates and opens a disk file for random disk read/writes.
LOOP	Controls the mesh cell indexing in the fluid-dynamics calculations.
KSTAR	Controls the indexing for the reactivity worth tables when the URANUS option is used.
LOAD	Los Alamos data loader.
SPLIT	A routine to plot N points from the X and Y arrays on a linear-linear, linear-logarithmic, logarithmic-linear, or logarithmic-logarithmic grid.
DLNLN	A routine to draw a linear-linear grid and put tick marks on the x and y axes.

TABLE N-I (CONT)

<u>Subroutine Name</u>	<u>Function of the Subroutine</u>
DLNLG	A routine to draw a linear-logarithmic grid.
DLGLN	A routine to draw a logarithmic-linear grid.
DLGLG	A routine to draw a logarithmic-logarithmic grid.
FRM	A routine to draw a frame around a picture.
MAXV	A routine to determine the maximum value of an array of real numbers.
MINV	Determines the minimum value of an array of real numbers.
ASCL	An automatic scaling routine.
DGAV	A routine to initialize the CJE07 common block.
SLLIN	A routine to scale the left or right grid linearly.
SBLIN	A routine to scale the top or botto grid linearly.
SBLOG	A routine to scale a grid logarithmically.
PLOT	A routine to plot an array of points. Each point may be represented by a symbol, and adjacent points may be connected by a solid line or vector.
XYZCON	A routine to calculate a viewpoint given a horizontal angle, a vertical angle, and the radius from the origin.
PICTUR	A routine to produce 3-D plots with hidden lines removed. A description of the input variables given in the prologue in the SIMMER-II source file.
INSECT	A routine to calculate the intersection of two lines.
CDRV	An internal routine used by PICTUR.
DCDRV	An internal routine used by PICTUR.
CADV	An internal routine used by PICTUR.

TABLE N-II
SIMMER-II COMMON DECK FUNCTIONS

<u>COMDECK Name</u>	<u>BLOCK Name</u>	<u>Function</u>
KOM	SIMCOM or blank	Contains the primary fluid-dynamics real variables.
KOM	INTCOM	Contains the primary fluid-dynamics integer variables.
TIME	TIME	Contains time steps, times, and time step control variables.
KPAS	PASS	Contains one row of fluid-dynamics variables for local calculations.
EQVR	None	Equivalences fluid-dynamics cell variables to the AASC array.
DIME	None	Dimensions fluid-dynamics cell variables.
REQV	None	Equivalences for parameter region data.
CONS	CONST	Contains all constants in the SIMMER-II system.
SET1	FWBGN1	Contains the problem name and LCM pointers.
SET2	FWBGN2	Contains maximum problem lengths and overlay file names.
SET5	LOCAL	Contains neutronics calculation control variables.
UN	UNITS	Contains file unit numbers.
IA	SIMCOM or blank	Contains the primary neutronics real data.
IA	INTCOM	Contains the primary neutronics integer data.
COMI	COMINP	Contains problem dimension variables common to both the neutronics and fluid-dynamics calculations.
STIM	STIM	Contains execution times for various sections of the SIMMER-II code.
PSED	PSEDIT	Contains editing control variables.
DMPE	None	Equivalences variables for restart routines.

TABLE N-II (CONT)

<u>COMDECK Name</u>	<u>BLOCK Name</u>	<u>Function</u>
IBMC	None	Double-precision statement for IBM computers.
EOS	EOS	Passes EOS data between the EOS routine P4 and the fluid-dynamics routines XCHANJ and EXFLUD.
ASET	ALLSET	Input data not needed beyond the initialization stage of the run and only used in overlay (1,6).
KINE	KINET	Contains URANUS point-kinetics variables.
SLVE	SOLVE	Parameters used by subroutines IMFLUD and MATSOL.
IMPD	NITIAL	Passes fluid dynamics initial data between REGINP and EOSINP.
STAT	STAT	Storages for the STAT timing option.
S2DR	S2DIR	Tabular EOS data.
XCHC	RXCHA	Real parameters used by subroutines XCHANJ, FAM, PHASE, and TSHTR.
XCHC	IXCHA	Integer parameters used by subroutines XCHANJ, FAM, PHASE, and TSHTR.
MCPC	MCPC	Parameters used in the multicomponent phase model.
CJE7 PIC CDR DCDR DLXY	CJE07 PICTC CDR DCDR DLXY	Common blocks used by the plotting routines.
KLC	KLC	LCM space for film buffer on the Livermore Time-Sharing System.
KLC1	KLC1 KLC2 KLC3	LCM storage arrays.

TABLE N-III
OVERLAY ENTRY POINTS

<u>Overlay Control Program</u>	<u>Overlay Function</u>
OVL00	Main control overlay.
OVL10	Input and initializing control.
OLV11	Neutronics input.
OLV12	Cross-section input.
OVL13	Bondarenko factors and multigroup cross sections.
OVL14	S_n coefficients.
OVL15	Completion of neutronics input.
OVL16	Read fluid-dynamics controls.
OVL17	Initialize SIMMER-II code.
OVL18	Read fluid-dynamics mesh cell input and initialize cell variables.
OVL19	Storage size testing and editing.
OVL1A	Initialize the CONST common block.
OVL 20	Neutronics calculation control.
OVL21	Mesh cell cross-section calculation.
OVL22	Flux shape calculation.
OVL24	Quasi-static calculation control.
OVL30	Fluid-dynamics control.
OVL31	Exchange-function calculation.
OVL32	Postprocessing file writing.
OVL33	Fluid-dynamics editing.

TABLE N-III (CONT)

<u>Overlay Control Program</u>	<u>Overlay Function</u>
OVL34	Fluid-dynamics explicit calculation.
OVL35	Fluid-dynamics implicit calculation.
OVL36	Material density and temperature transformation from fluid dynamics to neutronics.
OVL37	URANUS point kinetics calculation.
OVL38	Velocity vector plots.
OVL39	One-dimensional plotting.
OVL3A	Contour plotting.
OVL3B	Three-dimensional perspective plotting.
OVL3C	Fluid-dynamics editing of temporary mesh cell values.
OVL40	Restart dump read/write control.
OVL41	Restart dump read.
OVL42	Restart dump write.
OVL50	Execution time edits.

REFERENCE

1. J. R. Ford, "Common Los Alamos Mathematical Software Compendium," Los Alamos National Laboratory informal report (April 1984).

APPENDIX O

AASC ARRAY EQUIVALENCES

TABLE O-I
FORTRAN VARIABLES EQUIVALENCED TO THE AASC CELL STORAGE BLOCK

<u>Location in AASC ARRAY</u>	<u>FORTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
1	ALPHG	α_G	Vapor volume fraction.
2	ALPHS	α_S	Total volume fraction of all structure components.
3	P	p	Pressure (Pa).
4	UG	U_G	Vapor-field radial velocity (m/s).
5	UL	U_L	Liquid-field radial velocity (m/s).
6	VG	V_G	Vapor-field axial velocity (m/s).
7	VL	V_L	Liquid-field axial velocity (m/s).
8	RDPDRG	G^{-1}	The derivative of a macroscopic density with respect to pressure for constant internal energy (kg/N·m). For two-phase flow,

$$G^{-1} = \left(\frac{\partial \bar{\rho}_G}{\partial p} \right)_{e=\text{const}}$$

For single-phase flow,

$$G^{-1} = \left(\frac{\partial \bar{\rho}_L}{\partial p} \right)_{e=\text{const}}$$

These derivatives are computed from the appropriate EOS.

TABLE O-1 (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
9	QG	Q_G	Total energy transfer to the vapor per unit volume as computed in overlay XCHANG (W/m^3).
	TKIJ	K_{GL}	The liquid-vapor momentum exchange function from the previous cycle ($kg/m^3 \cdot s$).
10	FAIL3	-	<p>Flag for can-wall failure.</p> <p>FAIL3 = 1, the can-wall model restrains radial motion, and intersubassembly heat transfer to can wall is evaluated.</p> <p>FAIL3 = 0, radial motion is permitted, and no intersubassembly heat transfer to can wall is evaluated.</p> <p>FAIL3 = -1, radial motion is permitted, and intersubassembly heat transfer to the can wall is evaluated.</p>
11	PAREG	-	Parameter region flag.
12	EN1	Q_{N1}	Internal energy production rate for fertile fuel (W/m^3). For the point kinetics or power vs time options, this is the initial rate. For the quasi-static method, this is the energy production rate given by the flux shape and does not include the flux amplitude.

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
	ROUG	$(\bar{\rho}_{G^U G})_{i+1/2, j}$	Vapor radial momentum density (kg/m ² ·s).
	AXMG(1)	$-\Gamma_{g1}$	Total mass-transfer rate per unit volume to fertile fuel va- por (kg/m ³ ·s).
13	EN2	Q_{N2}	Internal energy production rate for fissile fuel (W/m ³). See location 12 (EN1) for further explanation.
	ROUL	$(\bar{\rho}_{L^U L})_{i+1/2, j}$	Liquid radial momentum density (kg/m ² ·s).
	AXMG(2)	$-\Gamma_{g2}$	Total mass-transfer rate per unit volume to fissile fuel va- por (kg/m ³ ·s).
14	EN3	Q_{N3}	Internal energy production rate for steel (W/m ³). See location 12 (EN1) for further explana- tion.
	ROVG	$(\bar{\rho}_{G^V G})_{i, j+1/2}$	Vapor axial momentum density (kg/m ² ·s).
	AXMG(3)	$-\Gamma_{g3}$	Total mass-transfer rate per unit volume to steel vapor (kg/m ³ ·s).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
15	EN4	Q_{N4}	Internal energy production rate for sodium (W/m^3). See location 12 (EN1) for further explanation.
	ROVL	$(\bar{\rho}_L V_L)_{i,j+1/2}$	Liquid axial momentum density ($kg/m^2 \cdot s$).
	AXMG(4)	$-\Gamma_{g4}$	Total mass-transfer rate per unit volume to sodium vapor ($kg/m^3 \cdot s$).
16	EN5	Q_{N5}	Internal energy production rate for control material (W/m^3). See location 12 (EN1) for further explanation.
	AXMG(5)	$-\Gamma_{g5}$	Total mass-transfer rate per unit volume to control material vapor ($kg/m^3 \cdot s$).
	PALPHG	α_{GP}	Product of the vapor volume fraction and the initial end-of-cycle pressure estimate. Used to transfer this information between IMFLUD IJ loops.
17	TSATF	$T_{Sat,1}$	The final saturation temperature for fuel as evaluated in the vaporization-condensation model (K). Used to transfer information from XCHANJ to EDFLPR.

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
	AXMG(6)	$-\Gamma_{g6}$	Total mass-transfer rate to the vapor field component of fission gas ($\text{kg}/\text{m}^3 \cdot \text{s}$).
18	KIJ	K_{GL}	Liquid-vapor momentum exchange function ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	DP	Δp	The change in pressure during a pressure iteration (Pa).
19	ROGBR	$\bar{\rho}_G$	Sum of the vapor-field component macroscopic densities (kg/m^3).
20	ROLBR	$\bar{\rho}_L$	Sum of the liquid-field component macroscopic densities (kg/m^3).
21	CQ	-	Temporary storage for printed and plotted variables.
	ROGBRT	$\bar{\rho}_G^{k+1}$	Intermediate value of the end-of-time-step vapor total macroscopic density (kg/m^3).
22	ROLBRT	$\bar{\rho}_L^{k+1}$	Intermediate value of the end-of-time-step liquid total macroscopic density (kg/m^3).
	TSATS	$T_{\text{Sat},2}$	The saturation temperature for steel following execution of the vaporization-condensation model (K). Used to transfer information from XCHANJ to EDFLPR.

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
23	DGKLLT	$\Delta t(\Gamma_{LG} + \Gamma_{LS} + K_{GL} + K_{LS})_{j+1/2}$	The product of the current time step and the sum of the liquid vaporization rate, the liquid adhesion rate on structure, the vapor-liquid momentum-exchange function, and the liquid-structure momentum-exchange function averaged over the axial momentum cell (i,j+1/2)(kg/m ³).
	FIJL	K_{LS}	Liquid-structure momentum exchange function (kg/m ³ ·s).
24	DGKGLT	$\Delta t(\Gamma_{GL} + K_{GL})_{j+1/2}$	The product of the current time step and the sum of the vapor-condensation rate and the vapor-liquid momentum-exchange function, averaged over the axial momentum cell (i,j+1/2) (kg/m ³).
	TSATNA	$T_{Sat,3}$	The final saturation temperature for sodium from the vaporization-condensation model (K). Used to transfer information from XCHANJ to EDFLPR.
	ADHERZ	$2\Gamma_{LS,j+1/2}\Delta z_{j+1/2}$	Loss flux of mass from the liquid field to the structure field, summed over two axial cells (kg/m ² ·s).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
25	DGKLG	$\Delta t(\Gamma_{LG} + K_{GL})_{j+1/2}$	The product of the current time step and the sum of the liquid vaporization rate and the vapor-liquid momentum-exchange function, averaged over the axial momentum cell (i,j+1/2) (kg/m ³).
	TSATC	$T_{sat,4}$	The saturation temperature for control material following execution of the vaporization-condensation model (K). Used to transfer information from XCHANG to EDFLPR.
	ADHERR	$2\Gamma_{LS,i+1/2} \Delta r_{i+1/2}$	Loss flux of mass from the liquid field to the structure field, summed over two radial cells (kg/m ² ·s).
26	DGKGG	$\Delta t(\Gamma_{GL} + K_{GL} + K_{GS})_{j+1/2}$	The product of the current time step and the sum of the vapor-condensation rate, the vapor-liquid momentum-exchange function, and the vapor-structure momentum-exchange function averaged over the axial momentum cell (i,j+1/2)(kg/m ³).
	FIJG	K_{GS}	Vapor-structure momentum-exchange function (kg/m ³ ·s).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
27	SSLT	Γ_{Lt}	Liquid-field total mass-exchange rate ($\text{kg/m}^3 \cdot \text{s}$).
	DHSTOR	D_h	The cell hydraulic diameter (m). Used to transfer information from XCHANJ to EDFLPR.
	AXMLI	$-\Gamma_{L1}$	Total mass-transfer rate to the liquid fertile fuel ($\text{kg/m}^3 \cdot \text{s}$).
28	SSGT	Γ_G	Vapor-field total mass-exchange rate ($\text{kg/m}^3 \cdot \text{s}$).
	VLMGNS	$ V_{GL}^{n+1} $	The difference between the liquid and vapor velocity used in the particle radius calculation (m/s). Used to transfer information from XCHANJ to EDFLPR.
29	RPSF	r_{psf}	Particle radius for solid fuel in the liquid field (m).
30	RPSS	r_{pss}	Particle radius for solid steel in the liquid field (m).
31	RSBR(1)	$\bar{\rho}_{s1}$	Macroscopic density of the solid fertile fuel in fabricated pellets (kg/m^3).
32	RSBR(2)	$\bar{\rho}_{s2}$	Macroscopic density of solid fissile fuel in fabricated pellets (kg/m^3).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
33	RSBR(3)	$\bar{\rho}_{s3}$	Macroscopic density of solid fertile fuel that has frozen in the structure field (kg/m ³).
34	RSBR(4)	$\bar{\rho}_{s4}$	Macroscopic density of solid fissile fuel that has frozen in the structure field (kg/m ³).
35	RSBR(5)	$\bar{\rho}_{s5}$	Macroscopic density of solid fuel pin cladding (kg/m ³).
36	RSBR(6)	$\bar{\rho}_{s6}$	Macroscopic density of sub-assembly can wall (kg/m ³).
37	RSBR(7)	$\bar{\rho}_{s7}$	Macroscopic density of solid control material (kg/m ³).
38	RSBR(8)	$\bar{\rho}_{s8}$	Macroscopic density of intra-granular fission gas in fabricated fuel (kg/m ³).
39	RSBR(9)	$\bar{\rho}_{s9}$	Macroscopic density of inter-granular fission gas in fabricated fuel (kg/m ³).
40	SIES(1)	e_{S1}	Specific internal energy of solid fabricated fuel (J/kg).
41	SIES(2)	e_{S2}	Specific internal energy of frozen fuel (J/kg).
42	SIES(3)	e_{S3}	Specific internal energy of solid cladding (J/kg).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
43	SIES(4)	e_{S4}	Specific internal energy of solid can wall (J/kg).
44	SIES(5)	e_{S5}	Specific internal energy of solid control material (J/kg).
45	FS(1)	Γ_{S1}	Mass-transfer rate per unit volume of fuel from fabricated fuel to liquid fuel ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	AXMLR(1)	$-\Gamma_{L2}$	Total mass-transfer rate to the liquid fissile fuel ($\text{kg}/\text{m}^3 \cdot \text{s}$).
46	FS(2)	Γ_{S2}	Mass-transfer rate per unit volume of fuel between frozen fuel in the structure field and liquid fuel ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	AXMLR(2)	$-\Gamma_{L3}$	Total mass-transfer rate to liquid steel ($\text{kg}/\text{m}^3 \cdot \text{s}$).
47	FS(3)	Γ_{S3}	Mass-transfer rate per unit volume of steel between cladding in the structure field and liquid steel ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	AXMLR(3)	$-\Gamma_{L4}$	Total mass-transfer rate to liquid sodium ($\text{kg}/\text{m}^3 \cdot \text{s}$).
48	FS(4)	Γ_{S4}	Mass-transfer rate per unit volume of steel between can wall in the structure field and liquid steel ($\text{kg}/\text{m}^3 \cdot \text{s}$).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
	AXMLR(4)	$-\Gamma_{L5}$	Total mass-transfer rate to liquid control material ($\text{kg}/\text{m}^3 \cdot \text{s}$).
49	FS(5)	Γ_{S5}	Mass-transfer rate per unit volume of control material between the structure field and the liquid field ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	AXMLR(5)	$-\Gamma_{L6}$	Total mass-transfer rate to fertile fuel particles in the liquid field ($\text{kg}/\text{m}^3 \cdot \text{s}$).
50	FS(6)	Γ_{S6}	Mass-transfer rate of intragranular fission gas to intergranular fission gas within fabricated fuel ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	AXMLR(6)	$-\Gamma_{L7}$	Total mass-transfer rate to fissile fuel particles in the liquid field ($\text{kg}/\text{m}^3 \cdot \text{s}$).
51	FS(7)	Γ_{S7}	Mass-transfer rate per unit volume of intergranular fission gas within fabricated fuel to the vapor field ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	AXMLR(7)	$-\Gamma_{L8}$	Total mass-transfer rate to steel particles in the liquid field ($\text{kg}/\text{m}^3 \cdot \text{s}$).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
52	FSC6	Γ_{SC}	Mass-transfer rate of fission gas from intragranular fission gas to intergranular fission gas for a heating rate of 450 K/s ($\text{kg/m}^3 \cdot \text{s}$).
53	RLBR(1)	$\bar{\rho}_{L1}$	Macroscopic density of liquid fertile fuel in the liquid field (kg/m^3).
54	RLBR(2)	$\bar{\rho}_{L2}$	Macroscopic density of liquid fissile fuel in the liquid field (kg/m^3).
55	RLBR(3)	$\bar{\rho}_{L3}$	Macroscopic density of liquid steel in the liquid field (kg/m^3).
56	RLBR(4)	$\bar{\rho}_{L4}$	Macroscopic density of liquid sodium in the liquid field (kg/m^3).
57	RLBR(5)	$\bar{\rho}_{L5}$	Macroscopic density of liquid control material in the liquid field (kg/m^3).
58	RLBR(6)	$\bar{\rho}_{L6}$	Macroscopic density of solid fertile fuel in the solid fuel particles in the liquid field (kg/m^3).
59	RLBR(7)	$\bar{\rho}_{L7}$	Macroscopic density of solid fissile fuel in the solid fuel particles in the liquid field (kg/m^3).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
60	RLBR(8)	$\bar{\rho}_{L8}$	Macroscopic density of solid steel particles in the liquid field (kg/m^3).
61	RLBRT(1)	$\bar{\rho}_{L1}$	Intermediate value of the advanced-time, liquid fertile fuel macroscopic density (kg/m^3).
		Q_{L1}	Total energy-transfer rate to the liquid fuel in the liquid field because of heat and mass-transfer (W/m^3).
62	RLBRT(2)	$\bar{\rho}_{L2}$	Intermediate value of the advanced-time, liquid fissile fuel macroscopic density (kg/m^3).
	QL(2)	Q_{L2}	Total energy-transfer rate to the liquid steel in the liquid field because of heat and mass-transfer (W/m^3).
63	RLBRT(3)	$\bar{\rho}_{L3}$	Intermediate value of the advanced-time liquid steel macroscopic density (kg/m^3).
	QL(3)	Q_{L3}	Total energy-transfer rate to the liquid sodium in the liquid field because of heat and mass-transfer (W/m^3).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
64	RLBRT(4)	$\bar{\rho}_{L4}$	Intermediate value of the advanced-time, liquid sodium, macroscopic density (kg/m ³).
	QL(4)	Q _{L4}	Total energy-transfer rate to the liquid control material in the liquid field from heat and mass-transfer (W/m ³).
65	RLBRT(5)	$\bar{\rho}_{L5}$	Intermediate value of the advanced-time liquid control material, macroscopic density (kg/m ³).
	QL(5)	Q _{L5}	Total energy-transfer rate to the solid fuel particles in the liquid field from heat and mass transfer (W/m ³).
66	RLBRT(6)	$\bar{\rho}_{L6}$	Intermediate value of the advanced-time macroscopic density for solid fertile fuel in solid fuel particles in the liquid field (kg/m ³).
	QL(6)	Q _{L6}	Total energy-transfer rate to the solid steel particles in the liquid field from heat and mass-transfer (W/m ³).
67	RLBRT(7)	$\bar{\rho}_{L7}$	Intermediate value of the advanced-time macroscopic density for solid fissile fuel in solid fuel particles in the liquid field (kg/m ³).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
	FL(1)	F_{L1}	Mass transfer per unit volume of fuel from solid fuel particles to liquid fuel ($\text{kg/m}^3 \cdot \text{s}$).
68	RLBRT(8)	$\bar{\rho}_{L8}$	Intermediate value of the advanced-time macroscopic density for solid steel particles in the liquid field (kg/m^3).
	FL(2)	F_{L2}	Mass-transfer rate per unit volume of steel from solid steel particles to liquid steel ($\text{kg/m}^3 \cdot \text{s}$).
69	SIEL(1)	e_{L1}	Specific internal energy of liquid fuel (J/kg).
70	SIEL(2)	e_{L2}	Specific internal energy of liquid steel (J/kg).
71	SIEL(3)	e_{L3}	Specific internal energy of liquid sodium (J/kg).
72	SIEL(4)	e_{L4}	Specific internal energy of liquid control material (J/kg).
73	SIEL(5)	e_{L5}	Specific internal energy of solid fuel particles in the liquid fuel (J/kg).
74	SIEL(6)	e_{L6}	Specific internal energy of solid steel particles in the liquid fuel (J/kg).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
75	GAMCE(1)	Γ_{G1L1}	Total mass-transfer rate per unit volume of fuel between liquid fuel and fuel vapor ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	DGKLLR	$\Delta t(\Gamma_{LG} + \Gamma_{LS} + K_{GL} + K_{LS})_{i+1/2}$	The product of the current time step and the sum of the liquid-vaporization rate, the liquid adhesion rate on structure, the vapor-liquid momentum-exchange function, and the liquid-structure momentum-exchange function averaged over radial momentum cell ($i+1/2, j$) (kg/m^3).
76	GAMC(2)	Γ_{G2L2}	Total mass-transfer rate of steel between liquid steel and steel vapor ($\text{kg}/\text{m}^3 \cdot \text{s}$).
	DGKGLR	$\Delta t(\Gamma_{GL} + K_{GL})_{i+1/2}$	The product of the current time step and the sum of the vapor-condensation rate and the vapor-liquid momentum-exchange function averaged over radial momentum cell ($i+1/2, j$) (kg/m^3).
77	GAMC(3)	Γ_{G3L3}	Total mass-transfer rate of sodium between liquid sodium and sodium vapor ($\text{kg}/\text{m}^3 \cdot \text{s}$).

TABLE O-1 (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
	DGKLGR	$\Delta t(\Gamma_{LG} + K_{GL})_{i+1/2}$	The product of the current time step and the sum of the liquid-vaporization rate and the vapor-liquid momentum-exchange function averaged over the radial momentum cell (i+1/2,j) (kg/m ³ ·s).
78	GAMCE(4)	Γ_{G4L4}	Total mass-transfer rate of control material between liquid control material and control material vapor (kg/m ³ ·s).
	DGKGGR	$\Delta t(\Gamma_{LG} + K_{GL} + K_{GS})_{i+1/2}$	The product of the current time step and the sum of the vapor-condensation rate, the vapor-liquid momentum-exchange function, and the vapor-structure momentum-exchange function averaged over the radial momentum cell (i+1/2,j)(kg/m ³).
79	RGBR(1)	$\bar{\rho}_{g1}$	Macroscopic density of fertile fuel vapor (kg/m ³).
80	RGBR(2)	$\bar{\rho}_{g2}$	Macroscopic density of fissile fuel vapor (kg/m ³).
81	RGBR(3)	$\bar{\rho}_{g3}$	Macroscopic density of steel vapor kg/m ³).
82	RGBR(4)	$\bar{\rho}_{g4}$	Macroscopic density of sodium vapor (kg/m ³).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
83	RGBR(5)	$\bar{\rho}_{g5}$	Macroscopic density of control material vapor (kg/m ³).
84	RGBR(6)	$\bar{\rho}_{g6}$	Macroscopic density of fission gas in the vapor field (kg/m ³).
85	RGBRT(1)	$\bar{\rho}_{g1}$	Intermediate value of the advanced-time, fertile fuel vapor macroscopic density (kg/m ³).
	QS(1)	Q_{S1}	Total energy-transfer rate to solid fabricated fuel from heat and mass-transfer (W/m ³).
86	RGBRT(2)	$\bar{\rho}_{g2}$	Intermediate value of the advanced-time, fissile fuel vapor macroscopic density (kg/m ³).
	QS(2)	Q_{S2}	Total energy-transfer rate to refrozen fuel in the structure field from heat and mass-transfer (W/m ³).
87	RGBRT(3)	$\bar{\rho}_{g3}$	Intermediate value of the advanced-time, steel vapor macroscopic density (kg/m ³).
	QS(3)	Q_{S3}	Total energy-transfer to cladding from heat and mass-transfer (W/m ³).

TABLE O-1 (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
88	RGBRT(4)	$\bar{\rho}_{g4}$	Intermediate value of the advanced-time, sodium vapor macroscopic density (kg/m^3).
	QS(4)	Q_{S4}	Total energy transfer rate to subassembly can wall because of heat and mass-transfer (W/m^3).
89	RGBRT(5)	$\bar{\rho}_{g5}$	Intermediate value of the advanced-time, control material vapor macroscopic density (kg/m^3).
	QS(5)	Q_{S5}	Total energy-transfer rate to solid control material from heat and mass-transfer (W/m^3).
90	RGBRT(6)	$\bar{\rho}_{g6}$	Intermediate value of the advanced-time macroscopic density for fission gas in the vapor field (kg/m^3).
91	SIEG	e_G	Specific internal energy of the vapor (J/kg).
			Between EXFLUD and IMFLUD, intermediate values of vapor energy per unit volume (J/m^3) are stored here.
92	QGWORK	$p_V \cdot \vec{v}$	Pressure-volume work term in the vapor energy equation (W/m^3).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
	TVN	T_G^{n+1}	The final vapor-field temperature from the vaporization-condensation model (K). Used to transfer information from XCHANJ to EDFLPR.
93	RPLF	r_{pL1}	Droplet radius for liquid fuel (m).
94	RPLS	r_{pL2}	Droplet radius for liquid steel (m).
95	RPLNA	r_{pL3}	Droplet radius for sodium (m).
96	RPLC	r_{pL4}	Droplet radius for liquid control material (m).
97	TEMGAS	T_G	Previous time-step vapor temperature for an initial guess for the equation-of-state iteration.
98	DHBELL	D_h^{-1}	Inverse hydraulic diameter used to compute particle jamming restrictions in EXFL2 (m^{-1}).
99	RHOLMI(1)	ρ_{L1}	The microscopic density for liquid fuel (kg/m^3).
100	RHOLMI(2)	ρ_{L2}	The microscopic density for liquid steel (kg/m^3).
101	RHOLMI(3)	ρ_{L3}	The microscopic density for liquid sodium (kg/m^3).

TABLE O-I (CONT)

<u>Location in AASC ARRAY</u>	<u>FORTTRAN Variable Name</u>	<u>Variable</u>	<u>Description (units)</u>
102	VELSOD	c_{L1}	The velocity of sound of liquid fuel (m/s).
103	ANOFI	$\alpha_{no,fl}$	No-flow filler volume fraction used in the plug calculative option.

APPENDIX P

DESCRIPTION OF NEUTRONICS COMMON ARRAYS

This appendix describes the contents of the IA and FIA arrays, which store integer and floating point data, respectively for the neutronics calculation. The integer data are described in Table P-I, and the floating point data are described in Table P-II. Neutronics LCM storage parameters are contained in labeled common block FWBGN1 and are described in Table P-III.

TABLE P-I

CONTENTS OF THE IA COMMON ARRAY

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
1	LETAIN	ETAIN	Input effective precursor concentrations for the transient neutronics initialization.
2	LBETIN	BETIN	Input effective delayed neutron fractions for the transient neutronics initialization.
3	ISN		Order of S_n . Used only if DIFF is off.
4	IGM		Number of neutron energy groups.
5	NRXS		Number of cross-section regions.
6	MT		Total number of materials.
7	LISMAX		Total number of cross-section types saved per group for each isotope.
8	MS		Number of mixture instructions.
9	LNISTP		Total length of cross-section block for each group.
10	LNISTP		Total number of input cross-section sets from ISOTXS file.
11			Not used.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
12	IGD		Number of delayed neutron precursor groups.
13	ITR		Flag for transient initialization option (0/1 = no/yes).
14	LBETAO	BETAO	Input physical delayed neutron fractions (used for transient neutronics initialization).
15	LMTTRN		Position of the transport cross section in the cell-wise macroscopic cross-section table.
16	LXSBUF	A(LXSBUF)	SCM buffer array used in cross-section input.
17	LISOXS	A(LISOXS)	SCM buffer array used to reorder cross-section input.
18	LFFBUF	A(LFFBUF)	SCM buffer array used in shielding factor input.
19	LISOFF	A(LISOFF)	Tabulated temperature and background shielding factor sets for all input isotopes for one group.
20	MM		$[ISN*(ISN+2)]/8$ = number of directions per octant.
21	IP		IM + 1.
22	JP		JM + 1.
23	IGP		IGM + 1.
24	ITJT		IT * JT.
25	LSIGP	A(LSIGP)	Tabulation points for background shielding factors for all input isotopes.
26	LPCNRM	PCNORM	Precursor normalization factors (used for transient neutronics initialization).
27	LTYFC	A(LTYFC)	Thermal yield per fission and per neutron capture. Not used if NEUPREP is defined.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
28	LSIGT	A(LSIGT)	Tabulation points for temperature shielding factors for all input isotopes.
29	NORDO		Number of initial unaccelerated outer (fission source) iterations.
30	LLOCA	A(LLOCA)	Record position number for each isotope on ISOTXS file.
31	LPCINT		Not used
32	LISDEN	A(LISDEN) or DENISO(LNISIP, IMJM)	Isotopic number densities for one cell and storage for isotopic inventory calculation if NEUPREP is off. Isotopic number densities for all cells of NEUPREP is on.
33	LMTNSF		Position of the product of the prompt fission yield and the fission cross section in the cell-wise macroscopic cross-section table.
34	LC	A(LC)	Microscopic cross sections for a neutron energy group. Not used if NEUPREP is defined.
35	LISLOC	A(LISLOC)	Storage for isotope record pointers used in cross-section input.
36	LLSIGT	NA(LLSIGT)	Pointers indicating the beginning of background shielding factor tabulation points for each isotope.
37	LA3	A(LA3)	Axial direction surface areas.
38	LA4	A(LA4)	Radial direction surface areas per unit height.
39	LA5	A(LA5)	Cell volume per unit height.
40	LQ	Q(LQ)	Source for the space-angle sweep.
41	LBR1	BR1(JT,MM)	Right boundary fluxes (in-down and out-down).
42	LBR2	BR2(JT,MM)	Right boundary fluxes (in-up and out-up).

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
43	LBT1	BT1(IT,MM)	Top boundary fluxes (in-down and in-up).
44	LBT2	BT2(IT,MM)	Top boundary fluxes (out-down and out-up).
45	LFL	A(LFL)	Cell-wise scalar flux for the current energy group.
46	LFISA	A(LFISA)	Cell-wise fission source.
47	LDC	NA(LDC)	Cross-section identification numbers for each component. These numbers assign a set of component mixtures to each cross-section block.
48	LXR	A(LXR)	Input radial coarse-mesh boundaries.
49	LYR	A(LYR)	Input axial coarse-mesh boundaries.
50	LYH	A(LYH)	Axial fine-mesh spacing.
51	LW	A(LW)	S_n angular quadrature weights.
52	LCM	A(LCM)	Radial direction cosines.
53	LCE	A(LCE)	Axial direction cosines.
54	LWM	A(LWM)	Products of quadrature weights and the radial direction cosines.
55	LWE	WETA(MM)	Products of quadrature weights and the axial direction cosines.
56	NORDM		Maximum order of Chebyshev polynomial used in outer iteration acceleration.
57	ICOS		Adjoint flux guess flag (0/1 = unity/cosine). Adjoint flux is the guess for the initial real flux solution.
58	LLMN	A(LLMN)	The number of isotopes used for each mixture.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
59	LMC	A(LMC)	Input mixture instructions.
60	LMD	A(LMD)	Input mixture densities.
61	LF	A(LF)	Cell-wise rebalance factors.
62	LFU	A(LFU)	Cell-wise upward partial currents.
63	LFD	A(LFD)	Cell-wise downward partial currents.
64	LFR	A(LFR)	Cell-wise rightward partial current.
65	LFLL	A(LFLL)	Cell-wise leftward partial currents.
66	LAB	A(LAB)	Cell-wise effective absorption removal rates.
67	LQQ	A(LQQ)	Cell-wise rebalance source.
68	LHA	A(LHA)	The main diagonal for solution of the transport rebalance equations.
69	LGA	A(LGA)	A "line" for the SLOR solution of rebalance factors.
70	LCHI	A(LCHI)	Input prompt fission spectrum.
71	IXSPTS		Pointwise cross-section print flag for initial state.
72	LVEL	A(LVEL)	Neutron energy group velocities.
73	LSOU	A(LSOU)	Total source in a group (same value as LQ).
74	LMTREM		Position of the removal cross section in the cell-wise macroscopic cross-section table.
75	LB1	A(LB1)	$1/\Delta z_j$.
76	LAL1	A(LAL1)	$\alpha_{m+1/2}/w_m$.
77	LAL2	A(LAL2)	$\alpha_{m-1/2}/w_m$.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
78	LALF	A(LALF)	Angular boundary fluxes.
79	LCTOT	A(LCTOT)	Cell-wise macroscopic cross section.
80	G		Number of the current neutron energy group.
81	OITNO		Current outer iteration number.
82	NN		ISN/2.
83	LPC	A(LPC)	Cell-wise precursor concentrations for the current delayed group.
84	LLAM	A(LLAM)	Delayed neutron precursor decay constants.
85	LBETA	A(LBETA)	Delayed neutron fractions.
86	LPS	A(LPS)	Delayed neutron emission spectra.
87	LLTEMT	NA(LLTEMT)	Pointers indicating the beginning of temperature-shielding factor tabulation points for each isotope.
88	IXSPTT		Pointwise cross-section print flag for the transient state.
89	LBINT	BINT(3,3,NDELAY)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).
90	LCBURN	CBURN(3,NDELAY)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).
91	LO	O(8,NDELAY,5)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).
92	LAQQ	AQQ(3,3)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).
93	LDEQ	DEQ(3)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
94	LEN	EN(2,3)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).
95	LSUMB	SUMB(3,3)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).
96	LSMRES	SUMREA(3)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).
97	LT	T(8,5)	Used in subroutine TSPK for amplitude equation solution; shares common with F(IT,JT).
98	LLC	A(LLC)	The amplitude-equation weighted delayed neutron precursor integral for the current time step.
99	LLCO	A(LLCO)	The amplitude-equation weighted delayed neutron precursor integral for the previous time step.
100	LSBETT	SBETT(IGD)	Used in subroutine TSPK for amplitude equation solution.
101	LREAT	REAT(3)	Used in subroutine TSPK for amplitude equation solution.
102	LVREAT	VREAT(3)	Used in subroutine TSPK for amplitude equation solution.
103	LBETCO	BETCO(3,IGD)	Used in subroutine TSPK for amplitude equation solution.
104	LBETVA	BETVA(3,IGD)	Used in subroutine TSPK for amplitude equation solution.
105	LBATA	BATA(IGD)	Used in subroutine TSPK for amplitude equation solution, shares common with F(IT,JT).
106	LPCUR	A(LPCUR)	Not used.
107	LDFISS	A(LDFISS)	Delayed neutron fission source integral for each delayed group.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
108	LPFLUX	A(LPFLUX)	Cell-wise flux for a previous time step; shares common with AB(IT,JT).
109	LAFLUX	A(LAFLUX)	Cell-wise adjoint flux; shares common with Q(IT,JT).
110	LEBETA	A(LEBETA)	Effective delayed neutron fractions.
111	LLFFGP	NA(LLFFGP)	Pointers indicating the beginning of shielding factors for each isotope for one group. Not used if NEUPREP is defined.
112	LNSIGT	NA(LNSIGT)	Number of background-shielding factor tabulation points for each isotope.
113	LNTEMT	NA(LNTEMT)	Number of temperature-shielding factor tabulation points for each isotope.
114	LNFFGP	NA(LNFFGP)	The number of shielding factors for each isotope for one group. Not used if NEUPREP is defined.
115	ISIGOD		Order of Lagrangian background-shielding factor interpolation. Not used if NEUPREP is defined.
116	LFFF	A(LFFF)	Shielding factors for all isotopes for one group. Not used if NEUPREP is on.
117	LMTMAX		Total number of types of cell-wise macroscopic cross sections.
118	LSIGO	A(LSIGO)	The background cross-section by isotope for one cell if NEUPREP is off. The logarithm of the background cross-section divided by number density per isotope for all cells if NEUPREP is on.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
119	LISTMP	A(LISTMP)	Temperatures by isotope for one cell if NEUPREP is off. Isotopic temperatures for all cells if NEUPREP is on.
120	LNMN	NA(LNMN)	Number of isotopes for each mixture.
121	ISHLD		Cross-section shielding defeat flag.
122	LISEI1		Position of the P1 elastic in-group scatter component in the input cross-section tables.
123	LISEO1		Position of the P1 elastic downscatter component in the input cross-section tables.
124	ITEMOD		Order of Lagrangian temperature-shielding factor interpolation. Not used if NEUPREP is defined.
125	LMTIGS		Position of the in-group scattering cross section in the cell-wise macroscopic cross-section table.
126	LMTDWN		Position of the first downscatter cross section in the cell-wise macroscopic cross-section table.
127	MAXTPT		Maximum number of temperature tabulation points for an input shielding factor set.
128	MAXPSO		Maximum number of background tabulation points for an input shielding factor set.
129	LISTOT		Position of the total cross-section in the input cross-section tables.
130	LISCAP		Position of the capture cross section in the input cross-section tables.
131	LPPK		First word of the amplitude equation parameter block.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
132	LISONM	A(LISONM)	Isotope names.
133	LDBETA	A(LDBETA)	The change in the effective delayed neutron fractions over the shape time step.
134	LPCS	A(LPCS)	Weighted delayed neutron precursor integrals for the beginning of the flux shape step.
135	LAH	A(LAH)	One of two factors to convert the fission source into the delayed neutron precursor source.
136	LBH	A(LBH)	One of two factors to convert the fission source into the delayed neutron precursor source.
137	LPFISS	A(LPFISS)	The cell-wise fission source for a previous time step; shares common with FL(ITP,JT).
138	LA6	A(LA6)	Δr .
139	LCD	A(LCD)	Component theoretical densities for each input component.
140	IREBLT		Total number of rebalance iterations for current energy group.
141	ITGAMM		Maximum number of gamma iterations permitted.
142	IREBAL		Total number of rebalance iterations for current inner iterations.
143	IRHOMP		Flag to control binary edit output to tape 6 and some printed edits.
144	LIFFF		Position of the fission shielding factor in the tables.
145	LRXSPT	A(LRXSPT)	Pointwise cross-section region numbers.
146	LISELO		Position of the elastic downscatter cross section in the input cross-section tables.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
147	NITOVR		Number of rebalance or flux inners between updates of the over-relaxation factor estimate.
148	LOMEG	A(LOMEG)	Over-relaxation factors for each group.
149	LISELI		Position of the elastic in-group scatter cross section in the input cross-section tables.
150	IOLYCS		Overlay indicator.
151	IMPJM		$(IM + 1) * JM$.
152	IMJMP		$IM * (JM + 1)$.
153	IMNN		$IM * NN$.
154	IMMM		$IM * MM$.
155	JMMM		$JM * MM$.
156	IMM1		$IM - 1$.
157	IMJM1		$IMJM - 1$.
158	LISREM		Position of the removal cross section in the input cross-section tables.
159	LISING		Position of the inelastic in-group cross section in the input cross-section tables.
160	LISFIS		Position of the fission cross section in the input cross-section tables.
161	LISMSC		Position of the miscellaneous cross section in the input cross-section tables.
162	LISNSF		Position of the $\nu\sigma_f$ cross section in the input cross-section tables.
163	LISTRN		Position of the transport cross section in the input cross-section tables.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
164	LISDWN		Position of the first downscatter cross section in the input cross-section tables.
165	LSIGRW	A(LSIGRW)	Temporary storage for the macroscopic elastic downscatter cross section. Not used if NEUPREP is on.
166	NSECT		LCM pointer or disk address of desired cell-wise cross-section components.
167			Not used.
168	LMTBLK		Length of macroscopic cell-wise cross-section block not including downscatter or power calculation components.
169	LCPDT	A(LCPDT)	Temporary storage for one row of component densities and temperatures. Not used if NEUPREP is on.
170	LA1	A(LA1)	$2\pi (r_{i+1} + r_i)$.
171	LA2	A(LA2)	$2\pi (r_{i+1} - r_i)$.
172	LIFFC		Position of the capture shielding factor in the tables.
173	LMTFDI	A(LMTFDI)	Position in the cell-wise cross-section tables of the diagonal elements of the five-band matrix used for the group diffusion solution.
174	LMTFCU	A(LMTFCU)	Position in the cell-wise cross-section tables of the outer diagonal elements of the five-band matrix used for the group diffusion solution.
175	LMTFCR	A(LMTFCR)	Position in the cell-wise cross-section tables of the super-diagonal elements of the five-band matrix used for the group diffusion solution.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
176	LSIGON	A(LSIGON)	Next iteration background cross section by isotope for one cell if NEUPREP is off. Background cross section is divided by number density per isotope for all cells if NEUPREP is on.
177	LFFCEL	A(LFFCEL)	Shielding factors by isotope for one cell if NEUPREP is off. Shielding factors by isotope for all cells if NEUPREP is on.
178	ITLMBG		Maximum number of iterations permitted over the background cross section.
179	LIFFE		Position of the elastic shielding factor in the background-, temperature-, and isotope-dependent set.
180	LIFFT		Position of the total shielding factor in the background-, temperature-, and isotope-dependent set.
181	LIFFB		Position of the last shielding factor in the background-, temperature-, and isotope-dependent set.
182	LTABTL	A(LTABTL)	Lowest tabulated value of temperature used for each isotope if NEUPREP is off. Lowest tabulated value of temperature for each isotope and each cell if NEUPREP is on.
183	LTABTH	A(LTABTH)	Highest tabulated value of temperature used for each isotope. Not used if NEUPREP is off.
184	LTABPH	A(LTABPH)	Highest tabulated value of background used for each isotope. Not used if NEUPREP is off.
185	LTABPL	A(LTABPL)	Lowest tabulated value of background used for each isotope. Not used if NEUPREP is off.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
186	LCELPT	NA(LCELPT)	Cells for which detailed shielding factor information is to be printed.
187	NCELPT		Number of cells for which detailed shielding factor information is to be printed.
188	LISCDN	A(LISCDN)	Isotopic number densities for each component for one cell if NEUPREP is off. Isotopic number densities for each component for all cells if NEUPREP is on.
189	LISIT1		Position of the P1 inelastic in-group scatter cross-section in the input cross-section tables.
190	LISIQ1		Position of the P1 inelastic out-group scatter cross section in the input cross-section tables.
191			Not used.
192			Not used.
193	GAMCON		Logical variable to indicate gamma iteration convergence in the quasi-static method.
194	KNTIN		Counter for the number of transport scattering source (inner) iterations for the current outer iteration for each group.
195	IFXUPD		A flag to indicate the status of the option to calculate an approximate flux shape at reactivity steps.
196	IFXUDL		An input variable indicating the maximum number of outer iterations permitted in the approximate flux shape update option. If zero, this option is not used.
197	IFXULX		A duplicate variable for IFXUDL; used to change IFXUDL at an appropriate point in the calculation if it is desired to change the value in a restart.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
198			Not used.
199			Not used.
200			Not used.
201			Not used.
202			Not used.
203			Not used.
204			Not used.
205			Not used.
206			Not used.
207	LISSWT	A(LISSWT)	Flag array for reordering temperature dependent shielding factors. Used for BROKXS input. Not used if NEUPREP is on.
208	LISSWP	A(LISSWP)	Flag array for reordering background-dependent shielding factors. Used for BRKOXS input. Not used if NEUPREP is on.
209	INVPRT		Flag for calculation and print of isotopic inventories.
210	INVREG		Number of isotopic inventory regions.
211	LINVRG		Inventory region boundary locations array.
212	LIFFD		Position of the elastic downscatter isotopic shielding factor in the table of calculated shielding factors.
213	LTCF1	A(LTCF1) or TCOEF1(2*ITJT, LNISIP)	Coefficients for linear interpolation of self-shielding factors over temperature for all isotopes for all cells. Used only if NEUPREP is on.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
214	LFFI1	A(LFFI1) or FFI1(MXITJT,MAXTPT)	Self-shielding factors interpolated over background for all needed temperatures for all isotopes for all cells. Used only if NEUPREP is on.
215	LVSIGN	A(LVSIGN) or VSIGN(ITJT,MAXTPT)	The "GATHERED" background cross sections for each isotope and each cell. Also used to temporarily store the total background cross section for all cells and the macroscopic elastic downscatter cross section for all cells. Used only if NEUPREP is defined.
216	LVSIGO	A(LVSIGO) or VSIGO(ITJT,MAXTPT)	The logarithm of the "GATHERED" background cross section for each isotope and each cell. Used only if NEUPREP is defined.
217	LIJIND	A(LIJIND) or IJIND(ITJT,MAXTPT)	A mapping to those cells containing a positive isotopic density for each temperature range for a given isotope. Used only if NEUPREP is defined.
218	LIJMID	A(LIJMID) or IJMIND(ITJT,LNISIP)	A mapping to those cells containing a positive isotopic density for each temperature range for each isotope. Used only if NEUPREP is defined.
219	LIJMCT	A(LIJMCT) or IJMCNT(MAXTPT,LNISIP)	The number of cells containing isotope at each temperature range. Used only if NEUPREP is defined.
220	LFFBF	A(LFFBF) or FFBUF(4*ITJT)	A temporary buffer array used to hold final interpolated shielding factors for all isotopes for all cells before "SCATTER"ing to permanent storage. Used only if NEUPREP is defined.
221	LCGFE	A(LCGFE) or CELGFE(ITJT)	The macroscopic elastic downscatter cross section for all cells for calculation of the downscatter source to the next energy group. Used only if NEUPREP is defined.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer of Array</u>	<u>Contents and Remarks</u>
222	LSIGOI	A(LSIGOI) or SIGOIS(ITJT, LNISIP)	The contribution of each isotope in each cell to the background cross section (that is, the isotopic total cross section). Also used to temporarily store the fractional changes in the background cross section for each isotope for each cell after each background iteration. Used only if NEUPREP is defined.
223	LTCF2	A(LTCF2) or TCOEF2(3*ITJT, LNISIP)	Coefficients for parabolic interpolation of self-shielding factors over temperature for all isotopes for all cells. Used only if NEUPREP is on.
224	LDLTR	A(LDLTR) or DLTR(3, ITJT)	The distances to the higher tabular points from the isotopic and cell-background cross sections. Needed for the B-spline background shielding factor interpolation. Used only if NEUPREP is defined.
225	LDELTL	A(LDELTL) or DELTL(3, ITJT)	The distances to the lower tabular points from the isotopic and cell-wise background cross sections. Needed for the B-spline background shielding factor interpolation. Used only if NEUPREP is defined.
226	LLIMT	A(LLIMT) or LIMT(MAXTPT)	The number of background-interpolated shielding factors needed for each temperature interpolation of a given isotope and reaction type. Used only if NEUPREP is defined.
227	LINDEX	A(LINDEX) or INDEX(4*ITJT)	A mapping between two storage schemes-, one contiguous and the other not contiguous. Used for "GATHER"ing or "SCATTER"ing. Used only if NEUPREP is defined.
228	LKTEML	A(LKTEML) or KTEMLL(LNISIP)	The lowest tabular temperature point needed over the entire spatial mesh for each isotope. Used only if NEUPREP is defined.

TABLE P-I (CONT)

<u>Position</u>	<u>Name</u>	<u>Pointer for Array</u>	<u>Contents and Remarks</u>
229	LKTEMH	A(LKTEMH) or KTEMHH(LNISIP)	The highest tabular temperature point needed over the entire spatial mesh for each isotope. Used only if NEUPREP is defined.
230	LILOC	A(LILOC) or ILOC(ITJT)	The first tabulation point below the value of background cross section for which B-spline interpolation is needed. Values are stored for each temperature set and all cells for which B-spline interpolation is needed for a given isotope. Used only if NEUPREP is defined.
231	LRGINV	A(LRGINV) or REGINV(INVREG*LNISIP)	The integrated number densities for each isotope in each inventory region. Used only if NEUPREP is defined.
232	IXSPFM		User-input flag to control the printing of performance statistics for the cross-section calculations. Used only if NEUPREP is defined.
233	LISECP		Position of the group-dependent isotopic capture energy in the input cross section tables. Used only if NEUPREP is on.
234	LISEFS		Position of the group-dependent isotopic fission energy in the input cross-section tables. Used only if NEUPREP is on.
235	ITEMIP		Input parameter to substitute parabolic temperature interpolation for B-spline temperature interpolation.
236			Not used.
237	MXITJT		MAXTEM * ITJT.
238-250			Not used.

TABLE P-II
CONTENTS OF THE FIA COMMON ARRAY

<u>Position</u>	<u>Name</u>	<u>Contents and Remarks</u>
1	POWER	Initial power.
2	OM	Inverse period.
3	RHOIN	Input initial reactivity for transient neutronics initialization.
4	GENTIN	Input initial generation time for neutronics transient initialization.
5	ZZ,Z	Radial geometry function used in FIXUP.
6	BB,B	Axial geometry function used in FIXUP.
7	CC,C	Angular geometry function used in FIXUP.
8	DD,D	Angular geometry function used in FIXUP.
9	T	Cell-centered flux used in FIXUP.
10	S,QT	Source used in FIXUP.
11	CT	Total cross section used in FIXUP.
12	SUMMU	$\sum_{M=1}^{MM} \text{COSMU}(M) * \text{WGT}(M).$
13	SUMETA	$\sum_{M=1}^{MM} \text{COSETA}(M) * \text{WGT}(M).$
14	AA	Radial geometric function used in FIXUP.
15	TI	Temporary i-flux used in FIXUP.
16	TJ	Temporary j-flux used in FIXUP.
17	TM	Temporary m-flux used in FIXUP.
18	EPSFAC	For the transport option, the ratio of the inner iteration convergence precision to the outer iteration convergence precision. For the diffusion option, the ratio of rebalance iteration convergence precision to the outer iteration convergence precision.

TABLE P-II (CONT)

<u>Position</u>	<u>Name</u>	<u>Contents and Remarks</u>
19	RVT	ITH/(VEL*DELTAT).
20	CURINS	Integral of the leakage reactivity evaluated at the beginning of a shape time step.
21	ORV	OM/VEL.
22	EPSBKG	Convergence precision for background iteration used on the shielding factor calculation.
23	GAMMAP	Previous value of gamma.
24	PINT	Power integral.
25	POM	Previous inverse period.
26	TPOW	Current power.
27	REACRT	Reactivity insertion rate.
28	CURINT	Current integral of the leakage reactivity for the neutron transport equation solution.
29	GAMMA	Current value of the quasistatic method constraint condition.
30	POWFS	Neutron flux amplitude.
31	GENTIM	Prompt neutron generation time.
32	REACT	Current value of the total reactivity.
33	TEBETA	Total effective delayed neutron fraction.
34	EPS1	Convergence precision for the amplitude equation solution.
35	PHI	Power amplitude.
36	PHIP	Previous power amplitude.
37	PINTGO	Normalized power integral error accumulated during the amplitude equation update.
38	POWFSP	Previous neutron flux amplitude.
39	POWDKP	Previous decay heat power amplitude.
40	POWDK	Decay heat power amplitude.

TABLE P-II (CONT)

<u>Position</u>	<u>Name</u>	<u>Contents and Remarks</u>
41	EPSPHI	Convergence precision for the amplitude equation solution.
42	PCURIN	Previous leakage contribution to the reactivity for the neutron transport solution.
43	QMULTP	Normalized power integral error accumulated during the reactivity update.
44	SUMN	Normalized power integral for the reactivity time step.
45	PINTG	Normalized power integral.
46	EIGEN	Eigenvalue of initial state ($EIGEN = 1.0/k_0$).
47	PHIPS	Power amplitude at the beginning of the flux-shape time step.
48	POWFS	Flux amplitude at the beginning of the flux-shape time step.
49	PINTPS	Normalized power integral at the beginning of the flux-shape time step.
50	REACTS	Reactivity at the beginning of the flux-shape time step.
51	GENTMS	Prompt neutron generation time at the beginning of the flux-shape time step.
53		Not used.
54	DTNSHP	Estimated flux-shape time step based on flux tilting.
55	PINTS	Normalized power integral for the reactivity time step (includes adjustment for the energy error accumulated in previous flux-shape updates).
56	OMR	Inverse period for the previous reactivity time step.
57	QMULTS	Normalized power integral for energy added to the system materials.
58	EPSPT	Convergence precision for mesh cell fission sources.

TABLE P-II (CONT)

<u>Position</u>	<u>Name</u>	<u>Contents and Remarks</u>
59	SIGO	Convergence precision for mesh cell fission sources.
60	VELFAC	Velocity factor for transient neutronics initialization.
61	EPSMIN	Minimum convergence precision for rebalance and inner iterations.
62	SPOWS	Power calculated from the flux shape (actual power divided by the amplitude function) at the beginning of the current flux-shape time step.
63	SPOW	Power calculated from the flux shape (actual power divided by the amplitude function) at current time.
64	FACRBL	For the neutron transport option, the ratio of rebalance iteration convergence precision to outer iteration convergence precision.
65	EPSRBL	For the transport option, the rebalance iteration convergence precision.
66	RFLMX	Reciprocal maximum initial cell flux.
67	ERRFXU	Integral flux-shape error reduction required for reactivity steps.
69	RAMPTO	Starting time for imposed reactivity ramps.
70	EPS10	Maximum deviation from unity permitted for GAMMAX before triggering a flux-shape time step.
71	EIGEN2	Eigenvalue guess for the real initial solution.
72-95		Not used.
96	OMEG	Overrelaxation factor for the current energy group.
97	RQZ, TEMP1	Temporary variable.
98	TEMP1, SUMA, TEMP	Temporary variable.
99	TEMP2, SUMB, FNORM	Temporary variable.

TABLE P-III
NEUTRONICS LCM STORAGE PARAMETERS

<u>LCM First Word Address</u>	<u>Length per Block</u>	<u>Contents</u>
100	TEMP3, RQ1	Temporary variable.
IPAF	ITJT*IGM	Adjoint flux.
IPFDCB	IT*IGM	Finite-difference coefficients for boundary. Present only if DIFF is defined.
IPFFAC	LNISIP*IGM*LIFFB *MAXTPT*MAXPSO	Input isotopic shielding factors.
IPFTAB	LNISIP*(MAXTPT+MAXPSO)	Tabulation points for shielding factors by isotope.
IPFX	ITJT*IGM	Neutron flux.
IPISXS	LNISTP*IGM	Input cross sections by isotope by group.
IPLFS1	ITJT	Previous fission source.
IPLFS2	ITJT	Previous fission source.
IPLKC	ITJT	Current leakage reactivity components. Used only if RHOMAP is on.
IPLKL	ITJT	Previous leakage reactivity components. Used only if RHOMAP is on.
IPPC	ITIJ*IGD	Delayed neutron precursor concentration.
IPPF	ITJT*IGM	Previous flux.
IPPK		Amplitude equation parameters by reactivity time step.
	(LIPSTP+2)*ITJT	If RHOMAP is on.
	ITJT	If RHOMAP is off.
IPQ	ITJT	Within-group scattering source. Used only if PIFF is off.

TABLE P-III (CONT)

<u>LCM First Word Address</u>	<u>Length per Block</u>	<u>Contents</u>
IPXSCT		Mesh-cell macroscopic cross sections by group.
	LMTMAX*ITJT	If COREXS is not defined.
	ITJT*((LMTBLK+NC)*IGM +(IGM*(IGM-1))/2)	If COREXS is defined.
ISHFLG	IGM*6*LNISIP	Shielding factor function type used for background interpolations. Present only if NEUPREP is defined.
ISWT	LNISIP	Tabular point at which the temperature interpolation for shielding factors becomes a linear extrapolation. Used only if NEUPREP is on.
ITPFLG	IGM*6*LNISIP	Shielding factor interpolation type used for temperature interpolations. Present only if NEUPREP is defined.

APPENDIX Q
SIMMER-II INPUT DATA

The data are separated into the following sections.

1. Problem control cards
2. Neutronics input for space-dependent kinetics (Skip if URANUS option is selected.)
3. Fluid-dynamics input
4. Point-kinetics input used for the URANUS option
5. Restart input

To aid the user in searching SIMMER-II input listings for data he desires to verify or change, identification cards precede certain sets of data. The information on these identification cards is read in a 18A4 format into dummy locations and is not used inside the code. These cards are to be used for alphanumeric descriptions of the input data that follow the cards.

For users on systems other than the Los Alamos LTSS/CTSS systems, the expression "edit on paper/film" should be interpreted as edit on TAPE9/TAPE12.

I. PROBLEM CONTROL INPUT

All runs, including restarts, must begin with the following card.

Card No. 1 (FORMAT: 2I6, 2A3, 6A10) ITS, ICAM, JHEAD, LABEL

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	ITS	Restart indicator. If ITS = 0, no restart; enter all data. If ITS ≠ 0, restart from the ABS(ITS) run and enter data changes through the NAMELIST format described in Sec. Q-V.
7-12	ICAM	Not used.
13-18	JHEAD	Not used.
19-78	LABEL	Film header label, up to 60 alphanumeric characters.

If ITS = 0, then the following cards must follow Card No. 1. If ITS ≠ 0, then a restart case is to be processed, and the input is as described in Sec. Q-V.

Card No. 2 (FORMAT: 1116) ITCARD, NEUSPR, NEULPR, NEUPLT, INTPLT, LPR, ISPR, IEDTSC, MNR36, ISK36, IDBFL

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	ITCARD	Number of title or job description cards to be entered.
7-12	NEUSPR	Neutronics short-edit output option. (Not required if using URANUS option.) NEUSPR = 0, no short edit; NEUSPR = 1, short edit on paper only; NEUSPR = 2, short edit on film only; NEUSPR = 3, short edit on paper and film.
13-18	NEULPR	Neutronics long-edit output option. (Not required if using URANUS option.) NEULPR = 0, no long edit; NEULPR = 1, long edit on paper only; NEULPR = 2, long edit on film only; NEULPR = 3, long edit on paper and film.
19-24	NEUPLT	Not used.
25-30	INTPLT	A postprocessing option. INTPLT = 0, data are not saved for postprocessing. INTPLT = 1, write time, power integral, inverse period, fluid-dynamics summations, and timing information for each overlay on output unit 6 to be postprocessed.
31-36	LPR	Fluid-dynamics long print control. LPR < 0, do not force neutronic update at long print time. The absolute value of LPR determines the device used for editing as indicated below. LPR > 0, force neutronic update at long print time: LPR = 1, long print on film only. LPR = 2, long print on film and printer. LPR = 3, long print on printer only. 10 < LPR < 99, contour and/or three-dimensional plots of the state variables and printing on film and/or printer. For the above values, LPR = L3DS*10 + LPRT, where LPRT = 1, film only; LPRT = 2, film and printer; LPRT = 3, printer only; and L3DS = 0, no plotting; L3DS = 1, contour plots of state variables; L3DS = 2, three-dimensional plots of the state variables; and

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
		L3DS = 3, contour and three-dimensional plots of state variables.
		LPR > 100, state variable three-dimensional movie option, where LPR = L3DM * 100 + L3DMVR and L3DMVR (≤ 103) is the position in the AASC array-of-state variable to be used in the movie. The first frame is repeated 100 times, and all other frames are repeated L3DM times. The use of the SIMMER-II postprocessor is recommended in lieu of on-line creation of movies.
37-42	ISPR	Short-print control indicator. The listing of summations of mass, momentum, and energy plus other conservation sums is provided each cycle if ISPR = 2 or only on those cycles that have a long print of cell data if ISPR = 1. If ISPR = 0, the short print is not edited.
43-48	IEDTSC	Fluid-dynamics time-step control print option. If IEDTSC = 0, no print of the ISTEP indicator or the time-step controls; if IEDTSC > 0, the above are printed each cycle.
49-54	MNR36	The maximum number of postprocessor files allowed on TAPE36.
55-60	ISK36	Postprocessor indicator. If ISK36 = 0, postprocessor files will be written. If ISK36 $\neq$ 0, postprocessor files will not be written.
61-66	IDBFL	If IDBFL > 0, print temporary variables in the AASC array after the execution of EXFLUD at the LPR time. This print will be for the cycle after the long print.

Card No. 3 (FORMAT: 18A4) (IDUSE (I) I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDUSE	Title or job description. ITCARD (see Card No. 2) Cards No. 3 have to be entered. The description on the first of these cards is printed as a header for the fluid-dynamics long prints.

Card No. 4 (FORMAT: 4E12.5) TWFIN, T2OMD, PARTL, TLJOB

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	TWFIN	Problem time when the calculation is to end (s).
13-24	T2OMD	Twenty-minute restart dump option. $T2OMD \leq 0.0$, force a restart dump every 60 min of CP time; $T2OMD > 0.0$, force a restart dump every T2OMD times 20 min.
25-36	PARTL	Fractional part of the machine time limit for the job to be used for SIMMER-II execution. The remainder of the time limit will be used to terminate the job. Default is 0.9.
37-48	TLJOB	If TLJOB $\neq 0$, TLJOB replaces the requested machine time limit. TLJOB must be less than the machine time limit. Only PARTL of TLJOB will be used, where TLJOB is in seconds.

Card No. 5 (FORMAT: 8I6) IBAR, JBAR, INPC(10), INPC(11), INPC(12), INPC(13), IM, JM

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	IBAR*	The number of mesh cells in the radial direction. Does not include fictitious boundary mesh cells.
7-12	JBAR*	The number of mesh cells in the axial direction. Does not include fictitious boundary mesh cells.

(The remaining values on this card are not needed if the URANUS option is defined.)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
13-18	INPC(10)	The farthest left fluid-dynamics mesh cell involved in the neutronics mesh.
19-24	INPC(11)	The farthest right fluid-dynamics mesh cell involved in the neutronics mesh.
25-30	INPC(12)	The lowest fluid-dynamics mesh cell involved in the neutronics mesh.
31-36	INPC(13)	The topmost fluid-dynamics mesh cell involved in the neutronics mesh.

*IBAR and JBAR are read and stored in variables INPC(1) and INPC(2).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
37-42	IM	The total number of neutronics mesh cells in the radial direction (including subdivisions of the fluid dynamics mesh).
43-48	JM	The total number of neutronics mesh cells in the axial direction (including subdivisions of the fluid dynamics mesh).

If an URANUS run is to be performed, the following section of input (space-dependent neutronics) is skipped, and the next card is Card No. 1 in Sec. Q-III.

II. SPACE-DEPENDENT KINETICS INPUT

Card No. 1 (FORMAT: 18A4) Identification Card.

Card No. 2 (FORMAT: 11I6) ISN, IGM, IGD, NORDO, NORDM, ITR, ICOS, MT, LNISIP, MS, NRXS

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	ISN	S_n order (not used if DIFF is defined). (Even integer only.) If negative, quadrature coefficients are taken from interface file SNCONS. Otherwise (for ISN = 2 through 16), built-in constants are used.
7-12	IGM	Number of neutron energy groups.
13-18	IGD	Number of delayed neutron precursor groups.
19-24	NORDO	Number of initial unaccelerated outer iterations to be performed before a sequence of Chebyshev accelerated iterations can begin. $NORDO \leq 3$. If NORDO is a large number, no acceleration is done (suggested value: 3).
25-30	NORDM	Largest order of Chebyshev accelerating polynomial permitted (suggested value: 10).
31-36	ITR	Transient initialization flag. If ITR = 1, the initial real flux calculation is based on transient conditions. If ITR = 0, the initial real flux calculation is based on stationary conditions.
37-42	ICOS	Initial flux guess flag. ICOS = 0, assume an initial iterate of unity for both real and adjoint flux. ICOS = 1, assume a cosine shape for the adjoint flux initial iterate and the adjoint solution for the real flux initial iterate. ICOS = 2, assume the adjoint flux initial iterate is input from TAPE29 and the adjoint solution is used for the real flux initial iterate.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
		ICOS = 3, assume an initial iterate of unity for the adjoint flux and input the real flux initial iterate from TAPE27.
		ICOS = 4, assume the adjoint flux initial iterate is input from TAPE29 and the real flux initial iterate is input from TAPE27.
43-48	MT	Total number of component materials created by the problem.
49-54	LNISIP	Number of input cross-section sets from the interface files ISOTXS and BRKOXS.
55-60	MS	Number of mixture instructions. See Sec. VI.B.4 and items A(LMC), and A(LCD) below.
61-66	NRXS	Number of cross-section regions.

Card No. 3 (FORMAT: 18A4) Identification Card for quasi-static method controls.

Card No. 4 (FORMAT: 6I6) KSTLIM, LIPSTP, ITGAMM, IWTF, IQUASI, IFXUDL

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	KSTLIM	Not used.
7-12	LIPSTP	Maximum allowed number of reactivity steps per shape step (suggested value: 7).
13-18	ITGAMM	Maximum number of gamma iterations permitted (suggested value: 5).
19-24	IWTF	Weighting function used for amplitude equation parameter integrations. If IWTF = 0, the stationary state adjoint flux is used. If IWTF = 1, unity is used (suggested value: 0).
25-30	IQUASI	Flag to override the quasi-static time step controls. IF IQUASI = 0, no effect; if IQUASI = 1, the flux-shape time step is always DTNMAX; if IQUASI = 2, the reactivity time step is always DTRMAX; if IQUASI = 3, both the flux shape and reactivity time steps are fixed.
31-36	IFXUDL	If the flux shape update method is used, maximum number of outer iterations permitted for flux shape updates per reactivity step (suggested value: 6). A positive value results in the use of the update method; a zero value results in the use of the flux-shape extrapolation approach.

Card No. 5 (FORMAT: 6E12.5) DELTAT, DELTR, DTNMAX, DTRMAX, EPSG, EPSPHI

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	DELTAT	Initial flux shape calculation time step (suggested value: 0.001).
13-24	DELTR	Initial reactivity time step (suggested value: 0.001).
25-36	DTNMAX	Maximum allowed shape step (suggested value: 0.01 if the extrapolation approach is used; 0.100 if the update approach is used).
37-48	DTRMAX	Maximum allowed reactivity step (suggested value: 0.001 if the extrapolation approach is used; 0.005 if the update approach is used).
49-60	EPSG	Gamma iteration convergence precision (suggested value: 0.001).
61-72	EPSPHI	Amplitude equation convergence precision (suggested value: 10^{-5}).

Card No. 6 (FORMAT: 6E12.5) EPS4, EPS17, EPS18, EPS6, EPS9, EPS10

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	EPS4	Quasistatic method time-step control reduction factor parameter (suggested value: 0.02).
13-24	EPS17	Quasistatic method time-step control exponential reduction factor (suggested value: 0.08516).
25-36	EPS18	Maximum permissible reactivity change per shape step (\$) (suggested value: 10.0).
37-48	EPS6	Maximum allowed relative change in the weighted current inner product per shape step (suggested value: 1.0).
49-60	EPS9	Maximum allowed relative spatial tilt in the total flux per shape step (suggested value: 1.0).
61-72	EPS10	Maximum deviation from unity permitted for the quasistatic constraint before taking a shape step (if IFXUDL > 0, the EPS10 control replaces the EPS18, EPS6, and EPS9 controls, whereas if IFXUDL = 0, EPS10 is inactive, and EPS18, EPS6, and EPS9 function as described).

Card No. 7 (FORMAT: 6E12.5) EPS11, EPS12, EPS13, EPS14, EPS15, EPS16

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	EPS11	Maximum allowed relative change in the total internal energy distribution of the fuel in a reactivity step (suggested value: 0.5).
13-24	EPS12	Maximum allowed relative change in the total internal energy distribution of the stainless steel in a reactivity step (suggested value: 0.5).
25-36	EPS13	Maximum allowed relative change in the total internal energy distribution of the sodium in a reactivity step (suggested value: 0.5).
37-48	EPS14	Maximum allowed relative change in the total density distribution of the fuel in a reactivity step (suggested value: 0.25).
49-60	EPS15	Maximum allowed relative change in the total density distribution of the stainless steel in a reactivity step (suggested value: 0.25).
61-72	EPS16	Maximum permissible relative change in the total density distribution of the sodium in a reactivity step (suggested value: 0.25).

Card No. 8 (FORMAT: 3E12.5) EPS5, EPS7, EPS8

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	EPS5	Maximum permissible reactivity change per reactivity step (\$) (suggested value: 5.0).
13-24	EPS7	Maximum number of reactivity steps per power decade (suggested value: 1.0).
25-36	EPS8	Maximum permissible inverse period change per reactivity step (1/s) (suggested value: 10^6).

Card No. 9 (FORMAT: 18A4) Identification card for edit controls.

Card No. 10 (FORMAT: 6I6) IXSPFM, IEDOPT, IXSPTS, IXSPTT, IEDXST, IRHOMP

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	IXSPFM	Prints statistics related to vectorization efficiency in the cross-section calculation. If IXSPFM > 0, print. If IXSPFM ≤ 0, no print. Used only if NEUPREP and XSSTAT1 are both defined.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
7-12	IEDOPT	Not used.
13-18	IXSPTS	Detailed print option for mesh-cell shielding-factor calculation. If IXSPTS = 1, print. If IXSPTS = 0, no print. If IXSPTS < 0, print specified cells.
19-24	IXSPTT	Transient-state detailed cross-section print option. Same as IXSPTS but for the transient state.
25-30	IEDXST	Cross-section print option. IEDXST = 0, no cross-section print IEDXST = 1, print input cross sections IEDXST = 2, include initial cell-wise cross-section print IEDXST = 3, include transient cell-wise cross-section print
31-36	IRHOMP	Mesh cell reactivity contributions flag. (IRHOMP is used only when RHOMAP is defined.) If INTPLT = 1, data will be saved on TAPE6 as indicated. IRHOMP = 0, no data save IRHOMP = 1, save reactivity contributions IRHOMP = 2, same as 1; in addition, save component densities and energies IRHOMP = 3, same as 2; in addition, save group fluxes. Note: Initial component densities and energies and group fluxes are saved when IRHOMP ≥ 1.

Card No. 11 (FORMAT: 2I6) INVPRT, INVREG

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	INVPRT	Inventory print flag INVPRT = 1, initial state only INVPRT = 2, transient only INVPRT = 3, both initial state and transient
7-12	INVREG	Number of inventory regions.

Card No. 12 (FORMAT: 18A4) Identification card for initial conditions.

Card No. 13 (FORMAT: 6E12.5) POWER, REACT, RAMPTO, RHOIN, GENTIN, OM

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	POWER	Initial power (W).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
13-24	REACRT	Programmed reactivity insertion rate ($\Delta k/s$). The neutron yield per fission is multiplied by $(1. + \text{REACRT} \cdot t)$ where t is the problem time.
25-36	RAMPTO	Time at which programmed ramp is to start.
37-48	RHOIN	Initial reactivity (dimensionless); used only if $\text{ITR} = 1$.
49-60	GENTIN	Initial generation time (s); used only if $\text{ITR} = 1$.
61-72	OM	Initial inverse period (1/s); used only if $\text{ITR} = 1$.

Card No. 14 (FORMAT: 18A4) Identification Card for convergence controls.

Card No. 15 (FORMAT: 4I6) ITLMOU, ITLMIN, IRBM, NITOVR

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	ITLMOU	Maximum number of outer iterations permitted (suggested value: 30).
7-12	ITLMIN	Transport option only (DIFF off). Maximum number of inner iterations per group permitted per outer iteration (suggested value: 20).
13-18	IRBM	For the transport option, the maximum number of rebalance iterations permitted per inner iteration. For the diffusion option, the maximum number of inner iterations per group permitted per outer iteration (suggested value: 50).
19-24	NITOVR	Number of rebalance iterations between successive over-relaxation (SOR) factor updates (suggested value: 5).

Card No. 16 (FORMAT: 6E12.5) EPS, EPSPT, EPSFAC, EPSMIN, FACRBL, ERRFXU

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	EPS	Convergence precision for the total fission source (suggested value: 10^{-6}).
13-24	EPSPT	Convergence precision for the pointwise fission source (suggested value: 10^{-5}).
25-36	EPSFAC	The factor relating the inner iteration convergence precision to the outer iteration convergence (suggested value: 75).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
37-48	EPSMIN	Minimum convergence precision for inner or rebalance iterations (suggested value: 10^{-8}).
49-60	FACRBL	Transport option only (DIFF off). The factor relating the rebalance iteration convergence precision to the inner iteration convergence precision (suggested value: 5).
61-72	ERRFXU	Convergence precision in the total fission source for the outer iteration flux shape updates at reactivity steps. Only used if IFXUDL > 0 (suggested value: 10^{-3}).

Card No. 17 (FORMAT: 2E12.5) RHOLIM, RHODTL

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	RHOLIM	Reactivity (\$) above which RHODTL limits the fluid-dynamics time step.
13-24	RHODTL	Maximum fluid-dynamics time step if reactivity exceeds RHOLIM.

Card No. 18 (FORMAT: 18A4) Identification card for shielding calculation controls.

Card No. 19 (FORMAT: 4I6) ISHLD, ISIGOD, ITEMOD, ITLMBG, ITEMIP

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	ISHLD	Flag to defeat cross-section shielding. If ISHLD = 0, shield input cross sections normally; if ISHLD = 1, no shielding is performed.
7-12	ISIGOD	Number of background points used by Lagrange interpolation of shielding factors (suggested value: 2). Used only if NEUPREP is not defined.
13-18	ITEMOD	Number of temperature points used by Lagrange interpolation of shielding factors (suggested value: 2). Used only if NEUPREP is not defined.
19-24	ITLMBG	Maximum number of iterations over background cross section permitted (suggested value: 1).
25-30	ITEMIP	Flag to alter selection of the interpolation function for the temperature dependence of the cross section self-shielding factors. If ITEMIP = 1, then all temperature interpolation flags with the value of 4 (B-spline) are converted to 3's (parabola). If ITEMIP $\neq$ 1, no change. Used only if NEUPREP is defined.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	EPSBKG	Maximum fractional change in background cross section permitted for any isotope (suggested value: 10^{-3}).

Except for the cross sections, the floating-point numbers and integers for the SIMMER-II space-dependent neutronics are read in a special read format such as in TWOTRAN-II.¹ These formats are [6(I1, I2, E9.4)] for reading floating-point numbers and [6(I1, I2, I9)] for integers. In each word of both of these formats, the first integer field, I1, designates the options listed in Table Q-I. The second integer field, I2, controls the execution of the option, and the remainder of the field, I9 or E9.4, is for the input data. All data blocks read with these formats must end with a 3 in the I1 field after the last word of the block. The identification cards in the section retain the 18A4 format.

TABLE Q-I

OPTIONS FOR SPECIAL READ FORMATS

<u>Value of I1</u>	<u>Nature of Option</u>
0 or blank	No action.
1	Repeat a data word in the 9 field the number of times indicated in the I2 field.
2	Interpolate linearly between the data word in the 9 field and the data word in the next field to obtain additional values for the number of words indicated in I2 field. <u>Not allowed for integers.</u>
3	Terminate reading of the data block. <u>A 3 must follow the last data word of each data block.</u>
4	Fill the remainder of the block with the data word in the 9 field. This operation must be followed by a terminate command (3).
5	Repeat the data word in the 9 field 10 times the value in the I2 field.
0 or blank	No action.
9	Skip to the next data card.

<u>Block Name and Dimension</u>	<u>Number of Entries</u>	<u>Comments</u>
IDCARD		Identification card for input cross-section names.
A(LISONM)	LNISIP+1	Names of materials for which cross sections and shielding factors are to be read. The last name denotes the cross-section input set from which the prompt fission spectrum is taken. Names must appear as in the data files.
IDCARD		Identification card for mesh data.
NCRAD		Number of neutronics mesh cells per fluid-dynamics mesh cell in the radial direction. The number of entries is equal to the number of fluid-dynamics mesh cells within the boundaries of the neutronics mesh.
NCAXI		Number of neutronics mesh cells per fluid-dynamics mesh cell in the axial direction. The number of entries is equal to the number of fluid-dynamics mesh cells within the boundaries of the neutronics mesh.
XR(IBAR)		Same as Card No. 9 in the Fluid-Dynamics Input Section.
YR(JBAR)		Same as Card No. 10 in the Fluid-Dynamics Input Section.
IDCARD		Identification card for cross-section blocks.
A(LDC)	NC*NRXS	Component cross-section identification numbers. These numbers assign a set of component mixtures to each cross-section block.
IDCARD		Identification card for component densities.
A(LCD)	MT	Theoretical densities (kg/m^3) for each component assigned previously.
IDCARD		Identification card for cross-section region boundaries.
RXSBD(4*NRXS)	4*NRXS	Cross-section region boundary cells. Enter by region, left boundary cell, right boundary cell, bottom boundary cell, and top boundary cell.
IDCARD		Identification card for mixture definitions.

<u>Block Name and Dimension</u>	<u>Number of Entries</u>	<u>Comments</u>
A(LMC)	MS	Numbers specifying the composition of component mixtures in terms of input materials. (The first name in A(LISONM) is number 1 and so on.) The series of numbers identifying each component always begins with a zero. Null compositions are identified by the zero with no subsequent entries.
IDCARD		Identification card for mixture number densities.
A(LMD)	MS	Atom number density (atoms/m ³) of the isotopes specified in A(LMC).
IDCARD		Identification card. Do not enter if IXSPTS ≥ 0 and IXSPTT ≥ 0.
A(LCELPT)	NCELPT*2	Mesh-cell locations for detailed cross-section edits, centered in pairs of numbers with the radial locations of all desired cells first and the axial locations second. The number of entries is twice the number of cells (given by the absolute value of IXSPTS and/or IXSPTT for IXSPTS*IXSPTT < 0). Do not enter if IXSPTS ≥ 0 and IXSPTT ≥ 0.
IDCARD		Identification card. Do not enter if INVREG = 0.
A(LINVRG)	INVREG*4	Region boundaries for isotopic inventory. Enter by region, bottom boundary cell, top boundary cell, left boundary cell, and right boundary cell. Note that this is a different order than for the cross-section region boundary cells. Also, regions can overlap.
IDCARD		Identification card. <u>Do not enter if IGD = 0.</u>
LAM(IGD)	IGD	Delayed neturon precursor decay constants (1/s). <u>Do not enter if IGD = 0.</u>
BETA(IGD+1)	IGD	Delayed neutron fission yield. <u>Do not enter if IGD = 0.</u>
PS(IGD,IGM)	IGD*IGM	Delayed neutron fission spectra. <u>Do not enter if IGD = 0.</u> Enter IGD values for the first energy group, then IGD values for the second, and so on.
IDCARD		Identification card. <u>Do not enter if ITR = 0 or IGD = 0.</u>
ETAIN(IGD)	IGD	Initial weighted precursor concentrations divided by the initial power. <u>Do not enter if IGD = 0 or ITR = 0.</u>

BETAIN(IGD)	IGD	Initial effective delayed neutron fraction. <u>Do not enter if IGD = 0 or ITR = 0.</u>
-------------	-----	--

III. FLUID-DYNAMICS INPUT

The input data for the fluid dynamics (as well as some input for the URANUS option) are listed in the order in which they are read into the code.

A. Integer Input

Card No. 1 (FORMAT: 18A4) (IDCARD (I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	IDCARD(I)	Message identifying fluid-dynamics integer parameter input.

Card No. 2 (FORMAT: 12I6) NOSET, JRIGID, IBOT, ITOP, NDT0, IVIS, NPAREG, IFGFAL, JCGV, NTMPIN, IRIGHT, LOCORF

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	NOSET	The number of mesh cell sets specified for the problem. Each mesh cell set defines the physical state in a rectangular area of mesh cells.
7-12	JRIGID	The number of mesh cells up the right boundary that are to have a rigid free-slip or rigid no-slip boundary condition. All mesh cells above the $J = JRIGID$ mesh cell are to have a continuative outflow boundary condition. $JRIGID \leq JBAR$. If $IRIGHT \leq -2$, the right boundary has the pressure specified rather than continuative outflow for $J > JRIGID$.
13-18	IBOT	Boundary condition indicator for the entire bottom boundary. ($IBOT \leq -2$ defines a time-dependent pressure boundary condition where the absolute value of IBOT defines the number of table entries in a pressure-vs-time table to be input, $IBOT = 0$ defines a rigid free-slip boundary condition, and $IBOT = 1$ defines a continuative outflow boundary condition.) $ IBOT \leq 23$.
19-24	ITOP	Boundary condition indicator for the entire top boundary. ($ITOP \leq -2$ defines a time-dependent pressure boundary condition where the absolute value of ITOP defines the number of entries in a pressure-vs-time table to be input, $ITOP = 0$ defines a rigid free-slip boundary condition, and $ITOP = 1$ defines a continuative outflow boundary condition.) $ ITOP \leq 23$.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
25-30	NDTO	The number of time steps for which the time step is maintained at less or equal the initial time-step size DTSTAE.
31-36	IVIS	Viscous shear indicator (IVIS = 0, no viscous shear; IVIS > 0 evaluate viscous shear).
37-42	NPAREG	The number of geometry parameter regions to be specified. $1 \leq \text{NPAREG} \leq 150$.
43-48	IFGFAL	Radial motion restraint condition indicator (IFGFAL = 0, unmodified; IFGFAL > 0, modified according to Card No. 7).
49-54	JCGV	The number of cells from the top of the problem to include the cover gas volume calculation for the long print. If JCGV = 0, then SCGV = 0 in the long print. (PLUG option off only.)
55-60	NTMPIN	Number of entries in table for sodium inlet temperature vs time. $0 \leq \text{NTMPIN} \leq 10$ (AEOS option on only).
61-66	IRIGHT	Boundary condition for the right-hand boundary above the JRIGID mesh cell. (IRIGHT ≤ -2 defines a time-dependent pressure boundary condition where the absolute value of IRIGHT defines the number of table entries in a pressure-vs-time table to be input, whereas IRIGHT ≥ 0 leaves the boundary as continuative outflow.) $ \text{IRIGHT} \leq 23$.
67-72	LOCORF	Axial orifice location. The orifice is located at the top of the node given by LOCORF. $0 \leq \text{LOCORF} \leq \text{JBAR}$. Note: If SIMMER-II has been updated with PLUG defined, JCGV is not read. The variables NTMPIN, IRIGHT, and LOCORF are moved to the left 6 spaces.

Card No. 3 (FORMAT 10I6) (NXCYC(N), N = 1, 10)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-60	NXCYC(N)	The number of time steps between long prints before time DTOC(N). IF NXCYC(N) = 0, then do long prints every DTO(N). (DTO(N) and DTOC(N) are defined in cards 17 and 18.)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
31-36	NITSRU	The number of matrix solution iterations in subroutine MATSOL (when the SLOR algorithm is used; MATSOL = ON) between matrix spectral radius and omega relaxation parameter updates. If the AIM algorithm is used (MATSOL = OFF), NITSRU is the relaxation parameter described in Appendix G. In this case, no relaxation is applied to the first NITSRU iterations; overall rebalance relaxation is applied the next NITSRU iterations and line rebalance relaxation is applied thereafter. The suggested value with both options is 5.
37-42	IORCO	Orifice coefficient input indicator (IORCO = 0, no orifice coefficients input; IORCO > 0, input orifice coefficients on Card No. 64).
43-48	IPOWER	Energy source option indicator. IPOWER < 0, no energy source; IPOWER = 0, the point kinetics model is used; IPOWER > 0, the user provides power vs time (required only if URANUS option is used).
49-54	IPREAN	The number of entries in the reactivity- or power-vs-time table (IPREAN = 0 when a driving reactivity of zero is to be used for IPOWER = 0 or a normalized power of unity is to be used for IPOWER > 0; $2 \leq \text{IPREAN} \leq 20$ tabular entries are to be specified). (Required only if URANUS option is used and not needed if IPOWER < 0.)
55-60	NDELAY	Number of delayed neutron groups. NDELAY ≤ 6 required only if URANUS option is used with IPOWER = 0.
61-66	NDKGRP	Number of decay heating groups NDKGRP ≤ 6 (required only if URANUS option is used with IPOWER = 0).
67-72	IGEOM	Horizontal geometry indicator (0 = slab, 1 = cylinder; not needed unless URANUS option is used).

If IFGFAL, Card No. 2, columns 43-48, is greater than zero, input Card No. 7.

Card No. 7 (FORMAT: 12I6) (IFAIL(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IFAIL(I)	Radial-motion and heat-transfer indicator for the right interface of mesh cell column I. IFAIL = 1, the can-wall model's radial motion restraint condition is applied, and intersub-assembly heat transfer to can wall is evaluated.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
31-36	NITSRU	The number of matrix solution iterations in subroutine MATSOL (when the SLOR algorithm is used; MATSOL = ON) between matrix spectral radius and omega relaxation parameter updates. If the AIM algorithm is used (MATSOL = OFF), NITSRU is the relaxation parameter described in Appendix G. In this case, no relaxation is applied to the first NITSRU iterations; overall rebalance relaxation is applied the next NITSRU iterations and line rebalance relaxation is applied thereafter. The suggested value with both options is 5.
37-42	IORCO	Orifice coefficient input indicator (IORCO = 0, no orifice coefficients input; IORCO > 0, input orifice coefficients on Card No. 64).
43-48	IPOWER	Energy source option indicator. IPOWER < 0, no energy source; IPOWER = 0, the point kinetics model is used; IPOWER > 0, the user provides power vs time (required only if URANUS option is used).
49-54	IPREAN	The number of entries in the reactivity- or power-vs-time table (IPREAN = 0 when a driving reactivity of zero is to be used for IPOWER = 0 or a normalized power of unity is to be used for IPOWER > 0; $2 \leq \text{IPREAN} \leq 20$ tabular entries are to be specified). (Required only if URANUS option is used and not needed if IPOWER < 0.)
55-60	NDELAY	Number of delayed neutron groups. NDELAY ≤ 6 required only if URANUS option is used with IPOWER = 0.
61-66	NDKGRP	Number of decay heating groups NDKGRP ≤ 6 (required only if URANUS option is used with IPOWER = 0).
67-72	IGEOM	Horizontal geometry indicator (0 = slab, 1 = cylinder; not needed unless URANUS option is used).

If IFGFAL, Card No. 2, columns 43-48, is greater than zero, input Card No. 7.

Card No. 7 (FORMAT: 12I6) (IFAIL(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IFAIL(I)	Radial-motion and heat-transfer indicator for the right interface of mesh cell column I. IFAIL = 1, the can-wall model's radial motion restraint condition is applied, and intersub-assembly heat transfer to can wall is evaluated.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
		IFAIL = 0, radial motion is permitted regardless of the can-wall density, and no intersub-assembly heat transfer to can wall is evaluated.
		IFAIL = -1, radial motion is permitted regardless of the can-wall density, and intersubassembly heat transfer to can wall is evaluated.

B. Problem Control Input

Card No. 8 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the fluid-dynamics problem control input.

Input as many Cards No. 9 as are needed to define all DR(I) for I = 1,2,...,IBAR. Initially the program sets MIJLAS = 0.

Card No. 9 (FORMAT: 3(E12.5,I12)) (DRZ(N), MIJ(N), N = 1,3)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	DRZ(1)	The radial mesh width (m) for DR(I). I = MIJLAS + 1, MIJLAS + 2,...,MIJ(1).
13-24	MIJ(1)	The rightmost radial mesh cell I value for which DR(I) = DRZ(1).
25-36	DRZ(2)	The radial mesh width (m) for DR(I), I = MIJ(1) + 1, MIJ(1) + 2,...,MIJ(2).
37-48	MIJ(2)	The rightmost radial mesh cell I value for which DR(I) = DRZ(2).
49-60	DRZ(3)	The radial mesh (m) for DR(I), I = MIJ(2) + 1, MIJ(2) + 2,...,MIJ(3).
61-72	MIJ(3)	The rightmost radial mesh cell I value for which DR(I) = DRZ(3).

Note that MIJ(1) < MIJ(2) < MIJ(3) unless MIJ(N) = 0 for N = 2 or 3 or both. If MIJ(1) < MIJ(2) < MIJ(3) < IBAR, the program sets MIJLAS = MIJ(3) and goes back and reads another Card No. 9. When MIJ(N) = IBAR for N = 1, 2, or 3, go to Card No. 10.

Input as many Cards No. 10 as needed to define all $DZ(N)$ for $J = 1, 2, \dots, JBAR$. Initially the program sets $MIJLAS = 0$.

Card No. 10 (FORMAT: 3(E12.5,I12)) (DRZ(N), MIJ(N), $N = 1, 3$)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	DRZ(1)	The axial mesh height (m) for $DZ(J)$, $J = MIJLAS + 1, MIJLAS + 2, \dots, MIJ(1)$.
13-24	MIJ(1)	The uppermost axial mesh cell J value for which $DZ(J) = DRZ(1)$.
25-36	DRZ(2)	The axial mesh height (m) for $DZ(J)$, $J = MIJ(1) + 1, MIJ(1) + 2, \dots, MIJ(2)$.
37-48	MIJ(2)	The uppermost axial mesh cell J value for which $DZ(J) = DRZ(2)$.
49-60	DRZ(3)	The axial mesh height (m) for $DZ(J)$, $J = MIJ(2) + 1, MIJ(2) + 2, \dots, MIJ(3)$.
61-72	MIJ(3)	The uppermost axial mesh cell J value for which $DZ(J) = DRZ(3)$.

Note that $MIJ(1) < MIJ(2) < MIJ(3)$ unless $MIJ(N) = 0$ for $N = 2$ or 3 or both. If $MIJ(1) < MIJ(2) < MIJ(3) < JBAR$, the program sets $MIJLAS = MIJ(3)$ and goes back and reads another Card No. 10. When $MIJ(N) = JBAR$ for $N = 1, 2$, or 3 , go to Card No. 11.

Card No. 11 (FORMAT: 4E12.5) AO, BO, G, ALPMIN

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	AO	A constant denoting the amount of full donor-cell weighting to be used (suggested value: 0.5).
13-24	BO	A constant denoting the amount of linearly interpolated convective fluxing to be used (suggested value: 0.0).
25-36	G	Acceleration constant of gravity in the axial direction acting on vapor and liquid. (+/- = upward/downward acceleration) (m/s^2) (suggested value: -9.8).
37-48	ALPMIN	The minimum allowable liquid-field volume fraction (default and suggested value: 10^{-4}).

Card No. 12 (FORMAT: 6E12.5) EPS, EPC, EPIR, EPI, EPHASE, EVAPOR

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	EPS	Compressible convergence criterion ϵ_{G1} . See Sec. IV.E.6 (suggested values: $10^{-4} \leq EPS \leq 10^{-2}$).
13-24	EPC	Compressible convergence criterion ϵ_{G2} . See Sec. IV.E.6 (suggested values: $10^{-6} \leq EPC \leq 10^{-3}$).
25-36	EPIR	Single-phase convergence criterion ϵ_{L1} used to tighten the liquid iteration convergence. See Sec. IV.E.6. (suggested values: $10^{-4} \leq EPIR \leq 10^{-2}$).
37-48	EPI	Single-phase convergence criterion ϵ_{L2} . See Sec. IV.E.6 (suggested values: $10^{-4} \leq EPI \leq 10^{-3}$).
49-60	EPHASE	Convergence criterion for the continuity EOS iteration in the heat-conduction-limited phase-transition model (suggested value and default: 10^{-8}).
61-72	EVAPOR	Overall convergence criterion for the heat-conduction-limited phase-transition model (suggested value and default: 0.05).

Card No. 13 (FORMAT: 6E12.5) EPSP, EPSL, EPSV, ROG CUT, SCRC, CAEOS

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	EPSP	Matrix inversion convergence criterion for the pressure (suggested value: 10^{-6} to 10^{-10}).
13-24	EPSL	Matrix inversion convergence criterion for the liquid-component densities (suggested value: 10^{-6} to 10^{-10}).
25-36	EPSV	Matrix inversion convergence criterion for the vapor-component densities (suggested value: 10^{-6} to 10^{-10}).
37-48	ROG CUT	Vapor density below which the vapor energy change time step control is not calculated or the minimum vapor density used for calculating EOS vapor saturation properties (suggested value: $10^{-5} \geq ROG CUT \geq 10^{-10}$).
49-60	SCRC	Structure condensation rate coefficient (suggested value: 0.1).
61-72	CAEOS	Convergence precision for the analytic EOS (default and suggested value: 10^{-10}).

Card No. 14 (FORMAT: 6E12.5) ALPHO, ALDRG, ALCSCO, ROSFAL, ROSMLT, RSCLAD

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	ALPHO	A vapor fraction of the flow volume fraction below which single-phase calculations are performed (typical value: 0.05).
13-24	ALDRG	Structure volume fraction above which structure drag is infinite (typical value: 0.90; default: 0.99).
25-36	ALCSCO	Vapor volume fraction below which vapor condensation on the structure does not occur (typical value: 0.3).
37-48	ROSFAL	Structure failure density (kg/m^3). The can wall is presumed to fail allowing radial motion when the can-wall macroscopic density is less than ROSFAL (typical value: 100).
49-60	ROSMLT	Fraction of the available fuel or steel mass that can be melted or frozen by the nonequilibrium model in a time step (typical value: 0.1).
61-72	RSCLAD	The minimum cladding structure density (kg/m^3) for which the nonflow volume fraction is added to the structure volume fraction (typical value: 100).

C. Problem Edit Input

Card No. 15 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying parameters controlling the fluid-dynamics output edit.

Card No. 16 (FORMAT: 5E12.5) (COLOUR(N), N = 1, 5)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-60	COLOUR(N)	Five parameters used for color film option (computer-dependent).

The following three pairs of variables control long prints, postprocessing dumps, and restarts. Because the final non-zero time is set to 2*TWFIN, at least two non-zero entries should be given each pair of variables. If only one time interval is specified and this time interval is input as greater than 2*TWFIN, difficulties will be encountered.

Card No. 17 (FORMAT: 6E12.5) (DTO(N), N = 1, 10)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	DTO(N)	The time interval (s) between long prints before time DTOC(N). Ten values must be entered.

Card No. 18 (FORMAT: 6E12.5) (DTOC(N), N = 1, 10)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	DTOC(N)	The time (s) at which to change to DTO(N+1) as the time interval between long prints. Ten values must be entered.

Card No. 19 (FORMAT: 6E12.5) (DTOP(N), N = 1, 10)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	DTOP(N)	The time interval (s) between postprocessor dumps before time DTOCP(N). Ten values must be entered.

Card No. 20 (FORMAT: 6E12.5) (DTOCP(N), N = 1, 10)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	DTOCP(N)	The time (s) at which to change to DTOP(N+1) as the time interval between postprocessor dumps. Ten values must be entered.

NOTE: Cards No. 21 and 22 are needed only if the SPECDMP option is used.

Card No. 21 (FORMAT: 6E12.5) (DTDMP(N), N = 1, 10)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	DTDMP(N)	The time interval (s) between restart dumps before time TDMP(N). Ten values must be entered.

Card No. 22 (FORMAT: 6E12.5) (TDMP(N), N = 1, 10)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	TDMP(N)	The time (s) at which DTDMP(N) is changed. Ten values must be entered.

Card No. 23 (FORMAT: 6E12.5) (RLMOV(N), N = 1, NCLR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	RLMOV(N)	The maximum density change (kg/m^3) of liquid-field density component N between postprocessor dumps (suggested value is the default value of 10^{30}). NCLR = 8.

Card No. 24 (FORMAT: 6E12.5) (RSMOV(N), N = 1, NCSR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	RSMOV(N)	The maximum density change (kg/m^3) of structure-field density component N between postprocessor dumps (suggested value is the default value of 10^{30}). NCSR = 9.

Card No. 25 (FORMAT: 6E12.5) (ELMOV(N), N = 1, NCLE)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	ELMOV(N)	The maximum specific energy change (J/kg) of liquid-field energy component N between postprocessor dumps (suggested value is the default value of 10^{30}). NCLE = 6.

Card No. 26 (FORMAT: 5E12.5) (ESMOV(N), N = 1, NCSE)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-60	ESMOV(N)	The maximum specific energy change (J/kg) of structure-field energy component N between postprocessor dumps (suggested value is the default value of 10^{30}). NCSE = 5.

Card No. 27 (FORMAT: 2E12.5) VLGMOV, VLLMOV

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	VLGMOV	The maximum vapor displacement (m) between postprocessor dumps (suggested value is the default value of 10^{30}). No longer used in SIMMER-II.
13-24	VLLMOV	The maximum liquid displacement (m) between postprocessor dumps (suggested value is the default value of 10^{30}). No longer used in SIMMER-II.

Card No. 28 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the view point parameters for three-dimensional plots.

Card No. 29 (FORMAT: 3E12.5) XVUS, YVUS, ZVUS

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	XVUS	The scale factor for X-view position, default $(-\sqrt{3})$, where $XUV = XVUS * XR$.
13-24	YVUS	The scale factor for Y-view position, default (-1.0) , where $YVU = YVUS * YT$.
25-36	ZVUS	The scale factor for Z-view position, default (4.0) , where $ZVU = ZVUS$.

D. Time-Step Data

Card No. 30 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the time-step data that follow.

Card No. 31 (FORMAT: 4E12.5) T, DTSTAR, DTMIN, COURNT

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	T	The problem starting time (s) (typical value: 0.0).
13-25	DTSTAR	The initial time-step size (s). After time step NDTO, the value of DTSTAR is chosen automatically according to time-step controls (typical value: between 10^{-6} and 10^{-5}).
25-36	DTMIN	The minimum time step size allowed (s).
37-48	COURNT	Courant stability condition multiplier (suggested value: 0.3).

Card No. 32 (FORMAT: 6E12.5) DTMAX, DJMAX, DTMPM, DROJAY, DPOWMX, DREAMX

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	DTMAX	The maximum time step allowed (s).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
13-24	DJMAX	The maximum fractional change in the smeared vapor or liquid density caused by condensation during a time step (set internally when vapor convective outflow is occurring) (suggested value: 0.50).
25-36	DTMPM	Dummy variable entering into the energy-density-change time-step control. Must be entered as nonzero but otherwise will not affect problem execution.
37-49	DROJAY	The maximum fractional change in the smeared vapor density caused by vaporization during a time step (suggested value: 10.0).
49-60	DPOWMX	The maximum fractional change in the power during a time step (needed only when the URANUS option is used) (suggested value: 1.0).
61-72	DREAMX	The maximum change in reactivity (\$) during a time step (needed only when the URANUS option is used and IPOWER = 0) (suggested value: 1.0).

The following card is only read if the URANUS option is defined.

Card No. 33 (FORMAT: 6E12.5) POW, GAMFER, GAMSS, GAMCOL, GAMCTL, DOPCO

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	POW	The initial total power of the system (W).
13-24	GAMFER	The fraction of the initial power generated in the fertile fuel.
25-36	GAMSS	The fraction of the initial power generated in the steel.
37-48	GAMCOL	The fraction of the initial power generated in the sodium coolant.
49-60	GAMCTL	The fraction of the initial power generated in the control material.
61-72	DOPCO	The Doppler coefficient ($T \frac{dk}{dt}$).

E. Failure Parameters

Card No. 34 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying structure and solid failure parameters because of melting.

Card No. 35 (FORMAT: 6E12.5) (FFLR(N), N = 1,7)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	FFLR(N)	The fraction of the heat of fusion that the energy of each structure and solid particle energy component must exceed its solidus energy before failure of the energy component. ($0.0 \leq \text{FFLR}(N) < \text{XFLR}(N)$; $\text{FFLR}(N) < 1.0$ for $N > 4$.) N = 1, fabricated fuel N = 2, frozen fuel N = 3, clad N = 4, can wall N = 5, control N = 6, solid fuel particles N = 7, solid steel particles

Card No. 36 (FORMAT: 4E12.5) (XFLR(N), N = 1, 4)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-48	XFLR(N)	The fraction of the failed structure material that is liquid. ($1.0 - \text{XFLR}(N)$) becomes the fraction of failed structure material that is solid particles. ($\text{FFLR}(N) < \text{XFLR}(N) \leq 1.0$.) N = 1, fabricated fuel, N = 2, frozen fuel, N = 3, clad, and N = 4, can wall.

Card No. 37 (FORMAT: 6E12.5) (RFLR(N), N = 1, 6)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	RFLR(N)	The maximum mass-transfer rate of failed structure to the liquid or vapor field. N = 1, fabricated fuel N = 2, frozen fuel N = 3, cladding N = 4, can wall N = 5, control N = 6, intergranular fission gas

Card No. 38 (FORMAT: 3E12.5) (CFLR(N), N = 1, 3)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-36	CFLR(N)	The maximum mass-transfer rate of failed structure to the liquid or vapor field before clad failure.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
		N = 1, fabricated fuel
		N = 2, control
		N = 3, intergranular fission gas

Card No. 39 (FORMAT: I6) NYLD

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	NYLD	Number of tabular data points relating the yield stress to temperature for calculation of pressure induced can-wall failure. If NYLD is zero, the next two cards are skipped. NYLD $\leq$ 15.

Card No. 40 (FORMAT: 6E12.5) (YLDTMP(N), N = 1, NYLD)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	YLDTMP(N)	Tabular temperature points for the yield-stress interpolation (K). Use additional cards as needed to obtain NYLD values.

Card No. 41 (FORMAT: 6E12.5) (YLD SIG(N), N = 1, NYLD)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	YLD SIG(N)	Tabular yield stress points corresponding to the previously given temperatures (Pa). Use additional cards as needed to obtain NYLD values.

F. Equation-of-State Data

Cards 42-47 specify the material properties and EOS data for each of the materials: fuel, M = 1; steel, M = 2; sodium, M = 3; control, M = 4; and fission gas, M = 5. Although a complete set of these cards must be input for each material in the above order, the only variables used for fission gas (M = 5) are those influencing the vapor properties. Additional cards are required for fuel, steel, and sodium as indicated.

Card No. 42 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the material whose properties and EOS data are to be input.

Card No. 43 (FORMAT: 5E12.5) ROSE(M), CVS(M), TMLT(M), HFUS(M), THCONS(M)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	ROSE(M)	The solid microscopic density of material M (kg/m^3). This variable is not used in the present version of SIMMER-II.
13-24	CVS(M)	The solid constant-volume specific heat of material M ($\text{J/kg}\cdot\text{K}$).
25-36	TMLT(M)	The melting temperature of material M (K).
37-48	HFUS(M)	The heat of fusion of material M (J/kg).
49-60	THCONS(M)	The thermal conductivity of material M in the solid phase ($\text{W/m}\cdot\text{K}$).

Card No. 44 (FORMAT: 5E12.5) ROLE(M), CVL(M), SIG(M), THCONL(M), XMUL(M)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	ROLE(M)	The liquid microscopic density of material M (kg/m^3). Default for liquid temperatures below the solidus energy for materials 1, 2, and 3. Also furnishes the default value for ROGCR.
13-24	CVL(M)	The liquid-phase constant-volume specific heat of material M ($\text{J/kg}\cdot\text{K}$).
25-36	SIG(M)	The surface tension of material M (kg/s^2).
37-48	THCONL(M)	The thermal conductivity of material M in the liquid phase ($\text{W/m}\cdot\text{K}$).
49-60	XMUL(M)	The viscosity of material M in the liquid phase ($\text{Pa}\cdot\text{s}$).

Card No. 45 (FORMAT: 6E12.5) PSTAR(M), TSTAR(M), TSUP(M), HSTAR(M), TCRIT(M), ZETA(M)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	PSTAR(M)	The p^* parameter in the vapor pressure-temperature relation for material M (Pa).
13-24	TSTAR(M)	The T^* parameter in the vapor pressure-temperature relation for material M (K).
25-36	TSUP(M)	The superheat of material M (K).
37-48	HSSTAR(M)	The h^* parameter in the heat-of-vaporization equation for material M (J/kg).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
49-60	TCRIT(M)	The critical temperature of material M (K).
61-72	ZETA(M)	The exponent parameter in the heat-of-vaporization equation for material M.

Card No. 46 (FORMAT: 6E12.5) CVG(M), GAM(M), ATOM(M), ENCRIT(M), WTMOL(M), EPSK(M)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	CVG(M)	The vapor-phase constant-volume specific heat of material M (J/kg·K).
13-24	GAM(M)	The ratio of the constant-pressure specific heat to constant-volume specific heat for material M in the vapor phase.
25-36	ATOM(M)	The molecular diameter of vapor material M (Å) (suggested values: fuel, 4.4; steel, 1.64; sodium, 3.567; control, 1.46; fission gas, 4.047; nitrogen, 3.798).
37-48	ENCRIT(M)	The internal energy of material M at the critical point (J/kg). Default: ENCRIT(M) = max [HSTAR(M), TMLT(M) + CVS(M) + HFUS(M) + CVL(M) * (TCRIT(M) - TMLT(M))].
49-60	WTMOL(M)	The molecular weight of material M.
61-72	EPSK(M)	The molecular force constant (1/K) of material M in the vapor phase (suggested values: fuel, 6468; steel, 7700; sodium, 1375; control, 5472; fission gas, 231).

Card No. 47 (FORMAT: 4E12.5) ROGCRT(M), ROGP95(M), ROLCRT(M), APARA(M)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	ROGCRT(M)	The critical density (kg/m ³) of material M. [Default: 0.375*ROLE(M)].
13-24	ROGP95(M)	The saturated vapor density (kg/m ³) of material M when TSAT(M) = 0.95*TCRIT(M). [Default: 0.25*ROGCRT(M)]. A variable T95ROG(M) eventually is stored in this location. T95ROG(M) = 0.95*ROGP95(M)*TCRIT(M).
25-36	ROLCRT(M)	The liquid critical density (kg/m ³) of material M. If a constant liquid density is desired for consistency with previous versions of SIMMER, ROLCRT(M) should be set to RHOA1(M) from Card 48. [Default: ROGCRT(M)].
37-48	APARA(M)	The parameter used to increase the gas constant above the critical temperature. This is $a_{G,M}$ in Eq. (III-55). [Default: 10 ¹⁰].

Card No. 54 (FORMAT: 6E12.5) (SVEL(N), N = 1, NCLR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	SVEL(N)	The sonic velocity of liquid-field density component N (m/s). NCLR = 8. (Default: 2 000 m/s.)

G. Heat-Transfer Correlation Data

Card No. 55 (FORMAT: 18A4) (IDCARD (I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying heat-transfer correlation data to follow.

Card No. 56 (FORMAT: 6E12.5) RLL(1,2), RLL(1,3), RLL (1,4), RLL (1,5), RLL(1,6), RLL(2,3)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	RLL(1,2)	Heat-transfer coefficient multiplier for heat transfer between liquid fuel and liquid steel.
13-24	RLL(1,3)	Heat-transfer coefficient multiplier for heat transfer between liquid fuel and liquid sodium.
25-36	RLL(1,4)	Heat-transfer coefficient multiplier for heat transfer between liquid fuel and liquid control.
37-48	RLL(1,5)	Heat-transfer coefficient multiplier for heat transfer between liquid fuel and solid fuel particles.
49-60	RLL(1,6)	Heat-transfer coefficient multiplier for heat transfer between liquid fuel and solid steel particles.
61-72	RLL(2,3)	Heat-transfer coefficient multiplier for heat transfer between liquid steel and liquid sodium.

Card No. 57 (FORMAT: 6E12.5) RLL(2,4), RLL(2,5), RLL(2,6), RLL(3,4), RLL(3,5), RLL(3,6)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	RLL(2,4)	Heat-transfer coefficient multiplier for heat transfer between liquid steel and liquid control.
13-24	RLL(2,5)	Heat-transfer coefficient multiplier for heat transfer between liquid steel and solid fuel particles.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
25-36	RLL(2,6)	Heat-transfer coefficient multiplier for heat transfer between liquid steel and solid steel particles.
37-48	RLL(3,4)	Heat-transfer coefficient multiplier for heat transfer between liquid sodium and liquid control.
49-60	RLL(3,5)	Heat-transfer coefficient multiplier for heat transfer between liquid sodium and solid fuel particles.
61-72	RLL(3,6)	Heat-transfer coefficient multiplier for heat transfer between liquid sodium and solid steel particles.

Card No. 58 (FORMAT: 3E12.5) RLL(4,5), RLL(4,6), RLL(5,6)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	RLL(4,5)	Heat-transfer coefficient multiplier for heat transfer between liquid control and solid fuel particles.
13-24	RLL(4,6)	Heat-transfer coefficient multiplier for heat transfer between liquid control and solid steel particles.
25-36	RLL(5,6)	Heat-transfer coefficient multiplier for heat transfer between solid fuel particles and solid steel particles.

Card No. 59 (FORMAT: 4E12.5) (COREL(I,1), I = 1,4)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	COREL(1,1)	The coefficient for defining the <u>liquid fuel-structure</u> Nusselt number (suggested value: 0.023).
13-24	COREL(2,1)	The Reynolds-number exponent for defining the <u>liquid fuel-structure</u> Nusselt number (suggested value: 0.8).
25-36	COREL(3,1)	The Prandtl-number exponent for defining the <u>liquid fuel-structure</u> Nusselt number (suggested value: 0.4).
37-48	COREL(4,1)	The additive constant for the <u>liquid fuel-structure</u> Nusselt number (suggested value: 0.0).

Card No. 60 (FORMAT: 4E12.5) (COREL(I,2), I = 1,4)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	COREL(1,2)	The coefficient for defining the <u>liquid-steel to structure</u> Nusselt number (suggested value: 0.025).

Card No. 48 (FORMAT: 4E12.5) RHOA1(M), RHOA2(M), RHOA3(M), RHOA4(M)

[Note: This card only read for materials 1, 2, and 3, that is, fuel, steel, and sodium.]

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	RHOA1(M)	First coefficient for the density-vs-temperature equation at low temperature. Variable a_{1m} in Eq. (III-71). [Default: ROL(1) for $M = 1$. ROL(M+1) for $M > 1$. See Card No. 53 for the definition of ROL.]
13-24	RHOA2(M)	Second coefficient for the density-vs-temperature equation at low temperature. Variable a_{2m} in Eq. (III-71).
24-36	RHOA3(M)	Third coefficient for the density-vs-temperature equation at low temperature. Variable a_{3m} in Eq. (III-71).
37-48	RHOA4(M)	Fourth coefficient for the density-vs-temperature equation at low temperature. Variable a_{4m} in Eq. (III-71).

Card No. 49 (FORMAT: 3E12.5) RHOB1(M), RHOB2(M), RHOB3(M)

[Note: This card is read only for materials 1, 2, and 3, that is, fuel, steel, and sodium.]

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	RHOB1(M)	First coefficient for the density-vs-temperature equation at high temperatures. Variable b_{1m} in Eq. (III-71).
13-24	RHOB2(M)	Exponent for the density-vs-temperature equation at high temperatures. Variable b_{2m} in Eq. (III-71).
25-36	RHOB3(M)	Last coefficient for the density-vs-temperature equation at high temperatures. Variable b_{3m} in Eq. (III-71).

Card No. 50 (FORMAT: 3E12.5) ASOUND, BSOUND, CSOUND

[Note: This card only read for material 1, fuel.]

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	ASOUND	Coefficient to compute velocity of sound as a function of temperature. Variable A in Eq. (III-64). [Default: 2212.339 m/s].
13-24	BSOUND	Exponent for the velocity of sound equation. Variable B in Eq. (III-64). [Default: 0.3539176].

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
25-36	CSOUND	Minimum fuel velocity of sound (m/s). Variable C in Eq. (III-64). [Default: 400 m/s].

Card No. 51 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying component properties to be input next.

Card No. 52 (FORMAT: 6E12.5) (ROS(N), N = 1, NCSR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	ROS(N)	The microscopic density of the solid phase of structure-field density component N (kg/m^3). NCSR = 9. N = 1, fabricated fertile fuel N = 2, fabricated fissile fuel N = 3, frozen fertile fuel N = 4, frozen fissile fuel N = 5, cladding N = 6, can wall N = 7, control N = 8, intragranular fission gas N = 9, intergranular fission gas

Card No. 53 (FORMAT: 6E12.5) (ROL(N), N = 1, NCLR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	ROL(N)	The microscopic density of liquid-field density component N (kg/m^3). NCLR = 8. N = 1, liquid fertile fuel N = 2, liquid fissile fuel N = 3, liquid steel N = 4, sodium N = 5, control N = 6, solid fertile N = 7, solid fissile fuel N = 8, solid steel

For liquid fuel, steel, and liquid sodium, the ROLs are to be used as default densities as indicated in Card No. 48. Also, they furnish the microscopic densities for liquid fuel and steel while melting structure or particles. For example, see Eqs. (III-5) and (III-6).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
13-24	COREL(2,2)	The Reynolds-number exponent for defining the <u>liquid-steel structure</u> Nusselt number (suggested value: 0.8).
25-36	COREL(3,2)	The Prandtl-number exponent for defining the <u>liquid-steel to structure</u> Nusselt number (suggested value: 0.8).
37-48	COREL(4,2)	The additive constant for the <u>liquid-steel to structure</u> Nusselt number (suggested value: 5.0).

Card No. 61 (FORMAT: 4E12.5) (COREL(I,3), I = 1,4)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	COREL(1,3)	The coefficient for defining the <u>liquid-sodium structure</u> Nusselt number (suggested value: 0.025).
13-24	COREL(2,3)	The Reynolds-number exponent for defining the <u>liquid-sodium structure</u> Nusselt number (suggested value: 0.8).
25-36	COREL(3,3)	The Prandtl-number exponent for defining the <u>liquid-sodium structure</u> Nusselt number (suggested value: 0.8).
37-48	COREL(4,3)	The additive constant for the <u>liquid-sodium structure</u> Nusselt number (suggested value: 5.0).

Card No. 62 (FORMAT: 4E12.5) (COREL(I,4), I = 1,4)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	COREL(1,4)	The coefficient for defining the <u>liquid-control structure</u> Nusselt number (suggested value: 0.023).
13-24	COREL(2,4)	The Reynolds-number exponent for defining the <u>liquid-control structure</u> Nusselt number (suggested value: 0.8).
25-36	COREL(3,4)	The Prandtl-number exponent for defining the <u>liquid-control structure</u> Nusselt number (suggested value: 0.4).
37-48	COREL(4,4)	The additive constant for the <u>liquid-control structure</u> Nusselt number (suggested value: 0.0).

Card No. 63 (FORMAT: 4E12.5) (COREL(I,5), I = 1,4)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	COREL(1,5)	The coefficient for defining the <u>vapor-structure</u> Nusselt number (suggested value: 0.023).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
13-24	COREL(2,5)	The Reynolds-number exponent for defining the <u>vapor-structure</u> Nusselt number (suggested value: 0.8).
25-36	COREL(3,5)	The Prandtl-number exponent for defining the <u>vapor-structure</u> Nusselt number (suggested value: 0.4).
37-48	COREL(4,5)	The additive constant for the <u>vapor-structure</u> Nusselt number (suggested value: 0.0).

Card No. 64 (FORMAT: 4E12.5) (COREL(I,6), I = 1,4)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	COREL(1,6)	The coefficient for defining the <u>vapor-liquid</u> Nusselt number (suggested value: 0.37).
13-24	COREL(2,6)	The Reynolds-number exponent for defining the <u>vapor-liquid</u> Nusselt number (suggested value: 0.6).
25-36	COREL(3,6)	The Prandtl-number exponent for defining the <u>vapor-liquid</u> Nusselt number (suggested value: 0.33).
37-48	COREL(4,6)	The additional additive constant for the <u>vapor-liquid</u> Nusselt number [suggested value: 0.0 because a 2.0 constant already exists in the correlation, Eq. (III-139)].

H. Drag Correlation Data

Card No. 65 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message describing drag correlation data that follow.

Card No. 66 (FORMAT: 5E12.5) CDR, WEBCRT, FNUG, FNUL, COAL

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	CDR	The drag coefficient used in the KACHINA drag correlation (suggested value: 1.0).
13-24	WEBCRT	Critical weber number (suggested value: 12.0).
25-36	FNUG	The vapor kinematic viscosity used in the viscous shear optional calculation (m^2/s).
37-48	FNUL	The liquid kinematic viscosity used in the viscous shear optional calculation (m^2/s).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
49-60	COAL	Coalescence coefficient for determining the particle droplet size (typical value: 1.0).

Card No. 67 (FORMAT: 6E12.5) PIEE, DSDM, TPF2, PARVIS, PKFMAX, ROGMXV

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	PIEE	Particle interference effect exponent that accounts for particle boundary layer effect in the liquid-vapor drag relation (suggested value: 2.5).
13-24	DSDM	Droplet size distribution multiplier. Provides a shift of droplet size to account for distribution effects (suggested value: 1.0).
25-36	TPWF2	Two-phase weighting factor. Divides the two-phase pressure drop between the liquid and vapor fields (suggested value: 0.5).
37-48	PARVIS	The particle viscosity coefficient multiplier (Pa·s) (default: 1.0).
48-60	PKFMAX	The maximum allowable particle packing fraction (default: 0.40).
61-72	ROGMXV	The maximum density transfer (kg/m^3) in a time step for which a one-degree difference between the saturation temperature and the liquid- or structure-field component temperature will determine a nonequilibrium vaporization or condensation rate. If the density transfer under such conditions exceeds ROGMXV, an equilibrium phase-transition rate will be calculated (currently overwritten by 10^{20}).

Card No. 68 (FORMAT: 6E12.5) FGS1, FGS2, FRGMIN, FLS1, FLS2, FRLMIN

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	FGS1	The coefficient of the expression defining the vapor-to-structure friction factor under turbulent flow conditions.
13-24	FGS2	The Reynolds number exponent in the expression defining the vapor-to-structure friction factor under turbulent flow conditions.
25-36	FRGMIN	Minimum vapor-structure friction factor (suggested value: 0.001).
37-48	FLS1	The coefficient of the expression defining the liquid-to-structure friction factor under turbulent flow conditions.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
49-60	FLS2	The Reynolds number exponent in the expression defining the liquid-to-structure friction factor under turbulent flow conditions.
61-72	FRLMIN	Minimum liquid-structure friction factor (suggested value: 0.001).

The turbulent flow vapor-to-structure friction factor is given by

$$f_{GS} = FGS1 * (Re_G)^{**}FGS2.$$

The turbulent flow liquid-to-structure friction factor is given by

$$f_{LS} = FLS1 * (Re_L)^{**}FLS2.$$

Two-phase friction factor multipliers are calculated as demonstrated in Sec. III.D.5.

I. Boundary Conditions

If ITOP (Card 2, word 4) $\geq$ 0, skip Cards 69, 70 and 71.

Card No. 69 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-78	IDCARD(I)	Message describing the time-dependent pressure for the top boundary condition that follows.

Card No. 70 (FORMAT: 6E12.5) (PTOP(I), I = 1, |ITOP|)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	PTOP(I)	The top boundary pressure (Pa) at time TPTOP(I) (Card 71). More than one card is needed if ITOP > 6 (Pa).

Card No. 71 (FORMAT: 6E12.5) (TPTOP(I), I = 1, |ITOP|)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	TPTOP(I)	The time (s) at which the top boundary pressure is PTOP(I) (Card 70). PTOP(I) and TPTOP(I) are to be ordered such that TPTOP(I) < TPTOP(I+1). More than one card is needed if ITOP > 6.

If IBOT (Card 2, word 3) ≥ 0 , skip Cards 72 and 73.

Card No. 72 (FORMAT: 6E12.5) (PBOT(I), I = 1, |IBOT|)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	PBOT(I)	The bottom boundary pressure (Pa) at time TPBOT(I) (Card 73). More than one card is needed if IBOT > 6.

Card No. 73 (FORMAT: 6E12.5) (TPBOT(I), I = 1, |IBOT|)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	TPBOT(I)	The specified time (s) at which the bottom boundary pressure is PBOT(I) (Card No. 72). PBOT(I) and TPBOT(I) are to be ordered such that TPBOT(I) < TPBOT(I+1). More than one card is needed if IBOT > 6.

If IRIGHT (Card 2, word 11) ≥ 0 , skip Cards 74 and 75.

Card No. 74 (FORMAT: 6E12.5) (PRIGHT(I), I = 1, |IRIGHT|)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	PRIGHT(I)	The right-side boundary pressure (Pa) at time TRIGHT(I) (Card No. 75). More than one card is needed if IRIGHT > 6.

Card No. 75 (FORMAT: 6E12.5) (TRIGHT(I), I = 1, |IRIGHT|)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	TRIGHT(I)	The specified time (s) at which the right-side boundary pressure is PRIGHT(I) (Card No. 74). PRIGHT(I) and TRIGHT(I) are to be ordered such that TRIGHT(I) < TRIGHT(I+1). More than one card is needed if IRIGHT > 6.

Insert Cards 76 and 77 if NTMPIN (Card 2, word 10) is other than zero (AEOS option only).

Card No. 76 (FORMAT: 6E12.5) (TMPIN(I), I = 1, NTMPIN)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	TMPIN	Inlet sodium temperature (K) at time TMPIN (Card No. 77). More than one card is required if NTMPIN > 6.

Card No. 77 (FORMAT: 6E12.5) (TTMPIN(I), I = 1, NTMPIN)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	TTMPIN	Time (s) at which the liquid sodium inlet temperature is TMPIN (Card No. 77). More than one card is required if NTMPIN > 6.

If IORCO (Card 6, word 7) $\leq$ 0, skip Card 78.

Card No. 78 (FORMAT: 6E12.5) (ORCO(I), I = 2, IP1)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	ORCO(I)	The orifice coefficient used to augment the pressure drop at the axial location given by variable LOCORF (Card 2, word 12). The algorithm used to modify the structure-fluid exchange coefficients is given by Eqs. (III-175) to (III-180). Because IP1 = IBAR + 1, more than one card is required if IBAR > 6.

J. Region-Wise Input

Repeat Cards 79-84 NPAREG (Card 2, word 7) times to specify the geometry parameters for each of the IREG = 1, NPAREG regions.

Card No. 79 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message describing region IREG.

Card No. 80 (FORMAT: 6E12.5) RGTYP(IREG), ALNOFL(IREG), PNOFL(IREG), RPELL(IREG), RCLAD(IREG), THCANW(IREG).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	RGTYP (IREG)	Geometric configuration indicator. Generally input as a positive number. A negative value is treated as explained in Table VI. RGTYP = 1, fuel pin channel with fuel pellets initially existing, RGTYP = 2, control rod channel with control material initially existing, RGTYP = 3, fuel pin channel without fuel pellets, RGTYP = 4, control rod channel without control pellets, RGTYP = 5, fuel pin channel without fuel pins,

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
RGTYPI = 6, control rod channel without control pins, and RGTYPI = 7, no structure in the cell.		
13-24	ALNOFL (IREG)	Non-structural volume fraction not available for flow in region IREG when the cladding density is above RSCLAD.
25-36	PNOFL (IREG)	Pressure (Pa) in the region IREG volume fraction not available for flow.
37-48	RPELL (IREG)	Outer radius (m) of the fuel pellets in region IREG.
49-60	RCLAD (IREG)	Outer radius (m) of the fuel pin cladding in region IREG. Not used in the current version of SIMMER-II.
61-72	THCANW (IREG)	Thickness (m) of the subassembly can wall in region IREG. Only used if NYLD (Card 39) is greater than zero.

Card No. 81 (FORMAT: 6E12.5) APELL(IREG), ACLAD(IREG), ACAN(IREG), VFPELL(IREG), VFCLAD(IREG), VFCANW(IREG).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	APELL (IREG)	Fuel pellet surface area per unit volume (1/m) in region IREG.
13-24	ACLAD (IREG)	Fuel pin cladding surface area per unit volume (1/m) in region IREG.
25-36	ACAN (IREG)	Can-wall surface area per unit volume (1/m) in region IREG.
37-48	VFPELL (IREG)	Initial volume fraction of the fuel pellets in region IREG.
49-60	VFCLAD (IREG)	Initial volume fraction of the cladding in region IREG.
61-72	VFCANW (IREG)	Initial volume fraction of the subassembly can wall in region IREG.

Card No. 82 (FORMAT: 6E12.5) DHINGE(IREG), DHNCLD(IREG), DHNPEL(IREG), HTCPEL(IREG), HTOCLD(IREG), HTCCAN(IREG).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	DHINGE (IREG)	Hydraulic diameter (m) for region IREG with intact geometry.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
13-24	DHNCLD (IREG)	Hydraulic diameter (m) for region IREG with fuel pellets having no cladding.
25-36	DHNPEL (IREG)	Hydraulic diameter (m) for region IREG with no fuel pins in the subassembly.
37-48	HTCPEL (IREG)	The heat-transfer coefficient ($W/m^2 \cdot K$) for the fuel pellets in region IREG.
49-60	HTOCLD (IREG)	The heat-transfer coefficient ($W/m^2 \cdot K$) for the cladding in region IREG.
61-72	HTCCAN (IREG)	The heat-transfer coefficient ($W/m^2 \cdot K$) for the subassembly can wall in region IREG.

Card No. 83 (FORMAT: 6E12.5) TSSF(IREG), FURSF(IREG), DPURSF(IREG),
RGRAIN(IREG), PERMF(IREG), RPMAX(IREG)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	TSSF (IREG)	The fuel temperature (K) that must be exceeded before fission gas release begins (suggested value: 1 950 K).
13-24	FURSF (IREG)	The fraction of the fabricated fuel that is unrestructured (typical value: 0.64).
25-36	DPURSF (IREG)	Pressure difference (Pa) across the unrestructured fuel annulus.
37-48	RGRAIN (IREG)	The effective radius (m) of the fuel grains in the unrestructured fuel (originally suggested value: 2.3×10^{-5}).
49-60	PERMF (IREG)	Fuel grain surface permeability (suggested value: $10^{-17} m^2$). A variable called FSC7 (IREG) eventually is stored in this location. FSC7 is a combination of constants and region-dependent variables used to evaluate Eq. (III-253).
61-72	RPMAX (IREG)	Maximum droplet radius allowed in region IREG (typical value: 0.001 m, default: 0.01 m).

Card No. 84 (FORMAT: 4E12.5) RPMIN(IREG), RHOSAT(IREG), ALNFCW(IREG),
GPFLG(IREG)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	RPMIN (IREG)	Minimum droplet radius allowed in region IREG (typical value: $5.E-5$).

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
13-24	RHOSAT (IREG)	Saturated grain boundary fission gas concentration in atoms per square meter of grain boundary surface (approximate range for values, 10^{19} - 10^{20} atoms/m ²). This variable is modified by RGRAIN, fission gas molecular weight, and Avogadro's number to actually store the saturated gas density in RHOSAT(IREG) when the code is executing.
25-36	ALNFCW (IREG)	Nonstructural volume fraction not available for flow that is associated with can wall. See Eq. (III-7) and Eq. (III-8) for the algorithm associated with ALNOFL(IREG) and ALNFCW(IREG).
37-48	GPFLG (IREG)	Parameter used in modeling to simulate fuel flow in intersubassembly gaps (default value: 1.0). If GPFLG is greater than 1.0, heat transfer to cladding and pellets is eliminated by setting HTCPEL, HTCCLD, APELL, and ACLAD to 10^{-10} .

K. Plug Data

Cards 85-88 are only to be input if PLUG has been defined.

Card No. 85 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying this data block.

Card No. 86 (FORMAT: 7I6) ISO, JSO, JSOT, JCGB, JCGT, ICGL, ICGR

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	ISO	Right-hand boundary of the plug
7-12	JSO	Bottom boundary of the plug
13-18	JSOT	Top boundary of the plug
19-24	JCGB	Bottom boundary of the cover gas region
25-30	JCGT	Top boundary of the cover gas region
31-36	ICGL	Left boundary of the cover gas region
37-42	ICGR	Right boundary of the cover gas region

Card No. 87 (FORMAT: 6I6) JFC, IFCL, IFCR, JFS, IFSL, IFSR

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	JFC	Node row for calculating core down-force
7-12	IFCL	Left boundary for core down-force calculation
13-18	IFCR	Right boundary for core down-force calculation
19-24	JFS	Node row for calculating annular shield down-force
25-30	IFSL	Left boundary for shield down-force calculation
31-36	IFSR	Right boundary for shield down-force calculation

Card No. 88 (FORMAT: 2E12.5) PLUGM, FPLUGB

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	PLUGM	Mass of the plug (kg)
13-24	FPLUGB	Force for plug breakaway (N)

L. Lower Boundary Velocities

Card No. 89 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the lower boundary initial velocities.

Card No. 90 (FORMAT: 6E12.5) (VG(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	VG(I)	The initial vapor axial velocity (m/s) at the bottom boundary for mesh cell column I. More than one card is needed if IBAR > 6.

Card No. 91 (FORMAT: 6E12.5) (VL(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	VL(I)	The initial liquid axial velocity (m/s) at the bottom boundary for mesh cell column I. More than one card is needed if IBAR > 6.

M. Mesh Point Sets

Repeat Cards 92-104 NOSET (Card 2, word 1) times to specify the mesh-cell parameters for each of the NOSET mesh-cell sets.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
		N = 1, fabricated fertile fuel
		N = 2, fabricated fissile fuel
		N = 3, frozen fertile fuel
		N = 4, frozen fissile fuel
		N = 5, cladding
		N = 6, subassembly can wall
		N = 7, control
		N = 8, intragranular fission gas
		N = 9, intergranular fission gas

Card No. 96 (FORMAT: 5E12.5) (TSI(N), N = 1, NCSE)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-60	TSI(N)	The initial temperature (K) of structure-field energy component N for this mesh cell set. NCSE = 5.
		N = 1, fabricated fuel
		N = 2, frozen fuel
		N = 3, cladding
		N = 4, subassembly can wall
		N = 5, control

Card No. 97 (FORMAT: 6E12.5) (RLBRI(N), N = 1, NCLR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	RLBRI(N)	The initial macroscopic density (kg/m ³) of liquid field density component N for this mesh-cell set. NCLR = 8.
		N = 1, liquid fertile fuel
		N = 2, liquid fissile fuel
		N = 3, liquid steel
		N = 4, liquid sodium
		N = 5, liquid control
		N = 6, solid fertile fuel particles
		N = 7, solid fissile fuel particles
		N = 8, solid steel particles

Card No. 98 (FORMAT: 6E12.5) (TLI(N), N = 1, NCLE)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	TLI(N)	The initial temperature (K) of liquid-field energy component N for this mesh-cell set. NCLE = 6.
		N = 1, liquid fuel
		N = 2, liquid steel
		N = 3, liquid sodium
		N = 4, liquid control
		N = 5, solid fuel particles
		N = 6, solid steel particles

If IINP $\neq$ 0 (Card 93, word 6), skip Card 99.

Card No. 99 (FORMAT: 6E12.5) PNI

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	PNI	The vapor pressure (Pa) for this mesh cell set.

If IINP $\leq$ 0 (Card 93, word 6), skip Cards 100 and 101.

Card No. 100 (FORMAT: 6E12.5) (ROGI(N), N = 1, NCGR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	ROGI(N)	The initial microscopic density (kg/m^3) of vapor-field density component N for this mesh-cell set. NCGR = 6. N = 1, fertile fuel vapor N = 2, fissile fuel vapor N = 3, steel vapor N = 4, sodium vapor N = 5, control vapor N = 6, fission gas

Card No. 101 (FORMAT: E12.5) TGI

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	TGI	The initial temperature (K) of the vapor mixture for this mesh-cell set.

Card No. 102 (FORMAT: 6E12.5) VGI, VLI, UGI, ULI, RPSFI, RPSSI

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	VGI	The initial vapor-field axial velocity (m/s) at the top interface of all mesh cells belonging to this mesh-cell set.
13-24	VLI	The initial liquid-field axial velocity (m/s) at the top interface of all mesh cells belonging to the mesh-cell set.
25-36	UGI	The initial vapor-field radial velocity (m/s) at the right interface of all mesh cells belonging to this mesh-cell set.
37-48	ULI	The initial liquid-field radial velocity (m/s) at the right interface of all mesh cells belonging to this mesh-cell set.
49-60	RPSFI	The radius (m) of the solid fuel particles.
61-72	RPSSI	The radius (m) of the solid steel particles.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-6	NB	The row number of the bottom row of mesh cells in this mesh-cell set.
7-12	NT	The row number of the top row of mesh cells in this mesh-cell set.
13-18	NL	The column number of the left column of mesh cells in this mesh-cell set.
19-24	NR	The column number of the right column of mesh cells in this mesh-cell set.
25-30	ICOMPO	<p>The number of the vapor material which is to be set in equilibrium with its liquid material when IINP = 0 (next parameter).</p> <p>ICOMPO = 1, fuel vapor ICOMPO = 2, steel vapor ICOMPO = 3, sodium vapor ICOMPO = 4, control vapor</p> <p>If IINP = 1, ICOMPO = 1 signals a mass-conservation calculation for all components in two-phase cells. If IINP = 1 and ICOMPO ≠ 1, no such mass conservation is attempted.</p>
31-36	IINP	Thermodynamic state input indicator. The options are as follows.
	IINP = -1	The liquid temperatures are specified and the vapor for material ICOMPO is put into equilibrium with the liquid material. For two-phase cells, the mass of material ICOMPO is conserved.
	IINP = 0	The vapor pressure for material ICOMPO is specified. The liquid temperature and vapor density of material ICOMPO then are adjusted to achieve equilibrium. Again, for two-phase cells, the mass of material ICOMPO is conserved.
	IINP = 1	The vapor temperature and microscopic vapor densities are specified. The cell pressure is computed from

$$p = \sum_{m=1}^{N_{MAT}} \rho_{Gm} c_{vgm} (\gamma_M - 1) T_G$$

The vapor density then is adjusted to conserve the cell pressure using the variable gas constant.
If ICOMPO = 1, mass conservation is attempted for all material components in two-phase cells.

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
	IINP = 2	The vapor temperature and microscopic vapor densities are specified. The gas constants, R_M , are calculated and the pressures are determined from

$$p = \sum_{m=1}^{N_{MAT}} \rho_{Gm} R_M T_G .$$

37-42	ISAT	The indicator to check if liquid-field energy component temperatures exceed their saturation temperatures (ISAT = 0, no action taken, ISAT > 0, the liquid component temperatures are checked against their saturation temperatures; if any are exceeded, a warning message is printed, the liquid-field energy component temperature is set equal to its saturation temperature, and its internal energy is recalculated accordingly).
43-48	ICELWS	The number of mesh-cell variables to have their values input for each mesh cell of this mesh-cell set.
49-54	IPAREG	The parameter region number assigned to this mesh-cell set.

If ICELWS $\leq$ 0 (Card 93, word 8), skip Card 94.

Card No. 94 (FORMAT: 12I6) (IFLAG(K), K = 1, ICELWS)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IFLAG(K)	The location number in the AASC storage array of the Kth parameter that is to have its value input for each mesh cell of this mesh-cell set. The AASC locations for internal energies are used to input the energy component temperatures, and the AASC locations for the vapor-field macroscopic densities are used to input the vapor-field microscopic densities. ($1 \leq \text{IFLAG}(K) \leq 103$). If ICELWS > 12, more than one card will be needed.

Card No. 95 (FORMAT: 6E12.5) (RSBRI(N), N = 1, NCSR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	RSBRI(N)	The initial macroscopic density (kg/m^3) of structure-field density component N for this mesh cell set. NCSR = 9.

If ICELWS = 0 (Card 93, word 8), skip Cards 103 and 104. Repeat Cards 103 and 104 ICELWS times to specify the individual mesh-cell values for each of the parameters identified by IFLAG.

Card No. 103 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the variable having its value specified for each mesh cell in this mesh-cell set.

Card No. 104 (FORMAT: 6E12.5) (AAROW(M), M = 1, NCELLS)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	AAROW(M)	A temporary array for reading the individual mesh-cell values for the IFLAG(K) location parameter in the AASC storage array. NCELLS = (NT - NB + 1) * (NR - NL + 1), the number of mesh cells in the current mesh-cell set. The order of value specification is by row, left to right, within an axial loop from the bottom row to the top row.

IV. POINT KINETICS INPUT

The following data parameters are to be specified when the URANUS option is used and IPOWER $\geq$ 0. If IPREAN = 0 (Card 6 of the fluid-dynamics input, word 9), skip Cards 1-3.

Card No. 1 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the time-dependent power or reactivity that follows.

Card No. 2 (FORMAT: 6E12.5) (PREATM(I), I = 1, IPREAN)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	PREATM(I)	The time (s) at which the driving reactivity (IPOWER = 0 Card 6, word 8 of the fluid-dynamics input) or normalized power (IPOWER > 0) is specified. When IPREAN > 6, more than one card is needed.

Card No. 3 (FORMAT: 6E12.5) (PREATB(I), I = 1, IPREAN)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	PREATB(I)	The driving reactivity (dollars) (if IPOWER = 0) or normalized power (if IPOWER > 0), at time PREATM(I). Linear variation of the reactivity or power in time between the specified time values is assumed. When IPREAN > 6, more than one card is needed.

Card No. 4 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the energy source spatial distribution that follows.

Card No. 5 (FORMAT: E12.5) PTTYPE

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	PTTYPE	Flag defining the type of energy source spatial distribution to be input (0.0 = specific power per unit mass, 1.0 = power density per unit volume).

Card No. 6 (FORMAT: 6E12.5) (PSHAPR(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	PSHAPR(I)	The unnormalized radial shape of the internal energy source. When IBAR > 6, more than one card is needed.

Card No. 7 (FORMAT: 6E12.5) (PSHAPZ(J), J = 1, JBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	PSHAPZ(J)	The unnormalized axial shape of the internal energy source. When JBAR > 6, more than one card is needed.

The product of PSHAPR(I) and PSHAPZ(J) defines the unnormalized internal energy source in mesh cell (I,J).

If IPOWER > 0 (Card 6, word 8 of the fluid-dynamics input), skip Cards 8-32.

Card No. 8 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the Doppler weight function.

Card No. 9 (FORMAT: 6E12.5) (WTDOPR(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTDOPR(I)	The unnormalized radial shape of the Doppler weight function. When IBAR > 6, more than one card is needed.

Card No. 10 (FORMAT: 6E12.5) (WTDOPZ(J), J = 1, JBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTDOPZ(J)	The unnormalized axial shape of the Doppler weight function. When JBAR > 6, more than one card is needed.

Card No. 11 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the fertile fuel reactivity worth distribution that follows.

Card No. 12 (FORMAT: 6E12.5) (WTFERR(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTFERR(I)	The radial shape of the fertile fuel reactivity worth. When IBAR > 6, more than one card is needed.

Card No. 12 (FORMAT: 6E12.5) (WTFERZ(J), J = 1, JBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTFERZ(J)	The axial shape of the fertile fuel reactivity worth. When JBAR > 6, more than one card is needed.

The product of WTFERR(I) and WTFERZ(J) defines the fertile fuel reactivity worth in mesh cell (I,J). One of these shapes should be entered as $\Delta k/kg \times 10^5$, and the other shape should be normalized to a maximum of 1.

Card No. 14 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the fissile fuel reactivity worth distribution that follows.

Card No. 15 (FORMAT: 6E12.5) (WTFISR(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTFISR(I)	The radial shape of the fissile fuel reactivity worth. When IBAR > 6, more than one card is needed.

Card No. 16 (FORMAT: 6E12.5) (WTFISZ(J), J = 1, JBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTFISZ(J)	The axial shape of the fissile fuel reactivity worth. When JBAR > 6, more than one card is needed.

The product of WTFISR(I) and WTFISZ(J) defines the fissile fuel reactivity worth in mesh cell (I,J). One of these shapes should be entered as $\Delta k/kg \times 10^5$, and the other shape should be normalized to a maximum of 1.

Card No. 17 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the steel reactivity worth distribution that follows.

Card No. 18 (FORMAT: 6E12.5) (WTSTRR(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTSTRR(I)	The radial shape of the steel reactivity worth. When IBAR > 6, more than one card is needed.

Card No. 19 (FORMAT: 6E12.5) (WTSTRZ(J), J = 1, JBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTSTRZ(J)	The axial shape of the steel reactivity worth. When JBAR > 6, more than one card is needed.

The product of WTSTRR(I) and WTSTRZ(J) defines the steel reactivity worth in mesh cell (I,J). One of these shapes should be entered as $\Delta k/kg \times 10^5$, and the other shape should be normalized to a maximum of 1.

Card No. 20 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the sodium coolant reactivity worth distribution that follows.

Card No. 21 (FORMAT: 6E12.5) (WTCOLR(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTCOLR(I)	The radial shape of the sodium coolant reactivity worth. When IBAR > 6, more than one card is needed.

Card No. 22 (FORMAT: 6E12.5) (WTCOLZ(J), J = 1, JBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTCOLZ(J)	The axial shape of the sodium coolant reactivity worth. When JBAR > 6, more than one card is needed.

The product of WTCOLR(I) and WTCOLZ(J) defines the sodium coolant reactivity worth in mesh cell (I,J). One of these shapes should be entered as $\Delta k/kg \times 10^5$, and the other shape should be normalized to a maximum of 1.

Card No. 23 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the control material reactivity worth distribution that follows.

Card No. 24 (FORMAT: 6E12.5) (WTCNTR(I), I = 1, IBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTCNTR(I)	The radial shape of the control material reactivity worth. When IBAR > 6, more than one card is needed.

Card No. 25 (FORMAT: 6E12.5) (WTCNTZ(J), J = 1, JBAR)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	WTCNTZ(J)	The axial shape of the control material reactivity worth. When JBAR > 6, more than one card is needed.

The product of WTCNTR(I) and WTCNTZ(J) defines the control material reactivity worth in mesh cell (I,J). One of these shapes should be entered as $\Delta k/kg \times 10^5$, and the other shape should be normalized to a maximum of 1.

Card No. 26 (FORMAT: 18A4) (IDCARD(I), I = 1, 18)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	IDCARD(I)	Message identifying the point kinetics data that follow.

Card No. 27 (FORMAT: 6E12.5) EPSPHI, EL, DKYLDS, REACIN

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-12	EPSPHI	The neutron flux convergence criterion (suggested values: $10^{-6} \leq \text{EPSPHI} \leq 10^{-5}$).
13-24	EL	The prompt neutron generation time.
25-36	DKYLDS	The total fraction of the power resulting from decay heating. (If DKYLDS = 0.0, the sum of the entries on Card 31 will be defined to be DKYLDS.)
37-48	REACIN	The initial reactivity of the system (dollars).

If NDELAY = 0 (Card 6, word 10, of the fluid-dynamics input), skip Cards 28-30.

Card No. 28 (FORMAT: 6E12.5) (EBETA(I), I = 1, NDELAY)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	EBETA(I)	The delayed neutron fraction in the delayed neutron precursor group I. (NDELAY $\leq$ 6).

Card No. 29 (FORMAT: 6E12.5) (ALAM(I), I = 1, NDELAY)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	ALAM(I)	The decay constant (1/s) for delay neutron precursor group I. (NDELAY $\leq$ 6).

If REACIN = 0.0 (Card 27, word 4), skip Card 30.

Card No. 30 (FORMAT: 6E12.5) (C(I), I = 1, NDELAY)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	C(I)	The initial concentration of delayed neutron precursor group I. (NDELAY $\leq$ 6).

If NDKGRP = 0 (Card 6, word 11, of the fluid-dynamics input), skip Cards 31 and 32.

Card No. 31 (FORMAT: 6E12.5) (DKYLD(I), I = 1, NDKGRP)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	DKYLD(I)	The decay heat fraction in decay heating group I. If DKYLD(1) = 0.0, the following default values are used. DKYLD(1) = 0.017890 DKYLD(2) = 0.023078 DKYLD(3) = 0.012820 DKYLD(4) = 0.006987 DKYLD(5) = 0.000939 DKYLD(6) = 0.003881

Card No. 32 (FORMAT: 6E12.5) (DKLAM(I), I = 1, NDKGRP)

<u>Columns</u>	<u>Variable</u>	<u>Description</u>
1-72	DKLAM(I)	The decay constant (1/s) for decay heating group I. If DKLAM(1) = 0.0, the following default values are used. DKLAM(1) = 0.19854 DKLAM(2) = 0.014366 DKLAM(3) = 0.0009362 DKLAM(4) = 7.620×10^{-5} DKLAM(5) = 9.725×10^{-6} DKLAM(6) = 8.875×10^{-7}

V. RESTART INPUT

All restart input begins with Card No. 1 of the problem control input data. The first word on the card, ITS, must be the restart number consistent with the restart file that provides the initial conditions for extending the calculation. When each restart file is written, the restart number is indicated on the SIMMER-II printed output file by the TAPE DUMP number.

If space-dependent neutronics problems are run, the first restart file occurs after the initial flux distributions are obtained. Thus, problem data can

be changed without recomputing the initial flux distribution. All other restart files are created during the transient part of the problem.

SIMMER-II allows certain input variables to be changed before a calculation is extended from a restart file. These data are read in after the first card by the FORTRAN namelist or free-formatted input. The use of NAMELIST is operating-system dependent, and the user should consult the operating system manuals before setting up restart input. Two sets of NAMELIST data are read. The first contains the fluid-dynamics and problem control data and is labeled HYDRST. This NAMELIST contains the problem finish time, TWFIN, which always must be redefined for a restart. The second NAMELIST is required only for space-dependent neutronics problems (URANUS option off) and is labeled NEURST.

The fluid-dynamics and problem control data contained in the HYDRST namelist include the following.

AO	EPSPHI (URANUS on)	OMP (not used)
BO	ESMOV	PARTL
ALCSCO	FGS1	PBOT
ALDRG	FGS2	PREATB (URANUS on)
ALPHO	FLS1	PREATM (URANUS on)
CAEOS (AEOS on)	FLS2	PRIGHT
CDR	IBOT	PTOP
COLOUR	IDBFL	RLL
COREL	INTPLT	RLMOV
COURNT	IPREAN (URANUS on)	ROGCUT
DJMAX	ISK36	ROSFAL
DPOWMX (URANUS on)	ISPR	ROSMLT
DREAMX (URANUS on)	ITOP	RSCLAD
DROJAY	ITOPT	RSMOV
DTDMP (SPECDCMP on)	LINE	T2OMD
DTMAX	LPR	TDMP (SPECDCMP on)
DTMIN	MITCRC	TLJOB
DTMPM	MITMAX	TMPIN (AEOS on)
DTO	MNR36	TPBOT
DTOC	NCYCPR	TPTOP
DTOCP	NITCRC	TRIGHT
DTOP	NITMAX	TTMPIN (AEOS on)
ELMOV	NITSRU	TWFIN
EPC	NODE	VLGMOV
EPI	NODES	VLLMOV
EPIR	NTMPIN (AEOS on)	WEBCRT
EPSKAC = EPS	NXCYC	

The variable NCYCPR can be used to specify the next time step number for which a fluid-dynamics long edit is desired. This parameter is supplemental to

the LPR, DTO, and DTOC parameters and allows the user to examine the problem conditions that could possibly be leading to code problems.

The neutronics data contained in the NEURST namelist include the following.

DELTAT	EPS16
DTNMAX	EPS17
DTR	EPS18
DTRMAX	ERRFXU
EPSG	FACRBL
EPSO	IEDXST
EPSFAC	IFXUDL
EPSMIN	IFXULX
EPSPHI	IQUASI
EPSPT	IRBM
EPS4	IRHOMP
EPS5	ITGAMM
EPS6	ITLMIN
EPS7	ITLMOU
EPS8	IXSPTT
EPS9	NEULPR
EPS10	NEUPLT
EPS11	NEUSPR
EPS12	NORDO
EPS13	NORDM
EPS14	RAMPTO
EPS15	REACRT

REFERENCE

1. K. D. Lathrop and F. W. Brinkley, "TWOTRAN-II: An Interfaced, Exportable, Version of the TWOTRAN Code for Two-Dimensional Transport," Los Alamos Scientific Laboratory report LA-4848-MS (July 1973).

APPENDIX R

CONSIDERATIONS ON ERRORS IN SHIELDING FACTOR CALCULATION RESULTING FROM COMMON ISOTOPES IN MATERIAL CROSS SECTIONS

MXS must be used in the NEUPREP-on option of SIMMER-II material cross sections created by the neutronics preprocessor. As noted in Chap. V, SIMMER-II will produce the same results (except for the effect of some minor approximations) from input based on individual isotopes as from input based on consistently defined materials, as long as the materials present in the same cells do not contain the same (common) isotopes. In this appendix, the common isotope error is examined, and recommendations for techniques to minimize or avoid it are made.

I. CAUSE AND CHARACTERISTICS OF MIXING ERROR

As seen in Chap. V, the mixing equations for the infinite cross sections and the self-shielding factors are formally exact (except for the small approximations of ν and χ) as long as the assumption that there are no common isotopes among materials is satisfied. Thus, the presence of common isotopes is the only possible cause of significant errors resulting from combining isotope cross sections.

The problem is illustrated below by considering a two-material system that has one common isotope.

Material A	• Isotope 1• Isotope 2
Material B	• Isotope 3• Isotope 2 (also referred to as 4 when convenient)

In this system, the equation relating the isotope σ_0^j and material σ_0^M , Eq. (V-16) is not valid. The exact solution for this system is

$$\sigma_0^1 = \frac{1}{N^1} [(N^2 + N^4)f^2(\sigma_0^{2+4})\sigma_t^2 + N^3f^3(\sigma_0^3)\sigma_t^3] \quad ,$$

$$\sigma_0^{2+4} = \frac{1}{N^2 + N^4} [N^1 f^1(\sigma_0^1) \sigma_t^1 + N^3 f^3(\sigma_0^3) \sigma_t^3] \quad , \quad \text{and} \quad (\text{R-1})$$

$$\sigma_0^3 = \frac{1}{N^3} [N^1 f^1(\sigma_0^1) \sigma_t^1 + (N^2 + N^4) f^2(\sigma_0^{2+4}) \sigma_t^2] \quad ,$$

where σ_0^{2+4} designates the exact background cross section of isotope 2 (and 4).

If we use Eq. (V-16) for this system, the resulting equations corresponding to Eqs. (R-1) are as follows.

$$\tilde{\sigma}_0^1 = \frac{1}{N^1} [N^2 f^2(\tilde{\sigma}_0^2) \sigma_t^2 + N^3 f^3(\tilde{\sigma}_0^3) \sigma_t^3 + N^4 f^2(\tilde{\sigma}_0^4) \sigma_t^2] \quad ,$$

$$\tilde{\sigma}_0^2 = \frac{1}{N^2} [N^1 f^1(\tilde{\sigma}_0^1) \sigma_t^1 + N^3 f^3(\tilde{\sigma}_0^3) \sigma_t^3 + N^4 f^2(\tilde{\sigma}_0^4) \sigma_t^2] \quad , \quad (\text{R-2})$$

$$\tilde{\sigma}_0^3 = \frac{1}{N^3} [N^1 f^1(\tilde{\sigma}_0^1) \sigma_t^1 + N^2 f^2(\tilde{\sigma}_0^2) \sigma_t^2 + N^4 f^2(\tilde{\sigma}_0^4) \sigma_t^2] \quad , \quad \text{and}$$

$$\tilde{\sigma}_0^4 = \frac{1}{4} [N^1 f^1(\tilde{\sigma}_0^1) \sigma_t^1 + N^2 f^2(\tilde{\sigma}_0^2) \sigma_t^2 + N^3 f^3(\tilde{\sigma}_0^3) \sigma_t^3] \quad ,$$

where the "tilde" quantities refer to values with mixing error included because of the common isotope.

The effective macroscopic cross section, both exact and with error, is expressed using Eqs. (R-1) and (R-2), respectively, as follows.

Exact solution

$$\Sigma_t = N^1 f^1 \sigma_t^1 + (N^2 + N^4) f^2 \sigma_t^{2+4} + N^3 f^3 \sigma_t^3 \quad (\text{R-3})$$

With-error solution

$$\tilde{\Sigma}_t = N^1 f^1(\tilde{\sigma}_0^1) \tilde{\sigma}_t^1 + N^2 f^2(\tilde{\sigma}_0^2) \tilde{\sigma}_t^2 + N^3 f^3(\tilde{\sigma}_0^3) \tilde{\sigma}_t^3 + N^4 f^4(\tilde{\sigma}_0^4) \tilde{\sigma}_t^4 \quad (\text{R-4})$$

The total error of Σ_t can be divided as follows.

$$E_{\text{total}} \equiv \tilde{\Sigma}_t - \Sigma_t, \text{ and} \quad (\text{R-5})$$

$$= E_1 + E_2 + E_3, \text{ where}$$

$$E_1 = N^1(f^1(\tilde{\sigma}_0^1)\tilde{\sigma}_t^1 - f^1(\sigma_0^1)\sigma_t^1),$$

$$E_2 = N^2(f^2(\tilde{\sigma}_0^2)\tilde{\sigma}_t^2 - f^2(\sigma_0^{2+4})\sigma_t^{2+4}) + N_4(f^4(\tilde{\sigma}_0^4)\tilde{\sigma}_t^4 - f^2(\sigma_0^{2+4})\sigma_t^{2+4}), \text{ and}$$

$$E_3 = N^3(f^3(\tilde{\sigma}_0^3)\tilde{\sigma}_t^3 - f^3(\sigma_0^3)\sigma_t^3). \quad (\text{R-6})$$

The partition of the error into E_1 , E_2 , and E_3 shows the error caused by isotope 1, 2, or (4) and 3, respectively. A first-order Taylor expansion is used to further expand each error term.

$$\begin{aligned} E_1 &= N^1\sigma_t^1 \frac{df^1}{d(\log\sigma_0)} \sigma_0^1 (\log\tilde{\sigma}_0^1 - \log\sigma_0^1) \\ &= N^1\sigma_t^1 \frac{df^1}{d(\log\sigma_0)} \sigma_0^1 \log\left[\frac{N^2f^2(\tilde{\sigma}_0^2)\sigma_t^2 + N^3f^3(\tilde{\sigma}_0^3)\sigma_t^3 + N^4f^2(\tilde{\sigma}_0^4)\sigma_t^2}{(N^2 + N^4)f^2(\sigma_0^{2+4})\sigma_t^2 + N^3f^3(\sigma_0^3)\sigma_t^3}\right], \\ E_2 &= N^2\sigma_t^2 \frac{df^2}{d(\log\sigma_0)} \sigma_0^{2+4} (\log\tilde{\sigma}_0^2 - \log\sigma_0^{2+4}) \\ &\quad + N^4\sigma_t^2 \frac{df_2}{d(\log\sigma_0)} \sigma_0^{2+4} (\log\tilde{\sigma}_0^4 - \log\sigma_0^{2+4}) \quad (\text{R-7}) \\ &= N^2\sigma_t^2 \frac{df^2}{d(\log\sigma_0)} \sigma_0^{2+4} \log\left[\left(1 + \frac{N^4}{N^2}\right) \frac{N^1f^1(\tilde{\sigma}_0^1)\sigma_t^1 + N^3f^3(\tilde{\sigma}_0^3)\sigma_t^3 + N^4f^2(\tilde{\sigma}_0^4)\sigma_t^2}{N^1f^1(\sigma_0^1)\sigma_t^1 + N^3f^3(\sigma_0^3)\sigma_t^3}\right] \\ &\quad + N^4\sigma_t^2 \frac{df_2}{d(\log\sigma_0)} \sigma_0^{2+4} \log\left[\left(1 + \frac{N^2}{N^4}\right) \frac{N^1f^1(\sigma_0^1)\sigma_t^1 + N^3f^3(\tilde{\sigma}_0^3)\sigma_t^3 + N^2f^2(\sigma_0^2)\sigma_t^2}{N^1f^1(\sigma_0^1)\sigma_t^1 + N^3f^3(\sigma_0^3)\sigma_t^3}\right], \end{aligned}$$

and

$$E_3 = N^3 \sigma_t^3 \frac{df^3}{d(\log \sigma_0)} \sigma_0^3 (\log \tilde{\sigma}_0^3 - \log \sigma_0^3)$$

$$= N^3 \sigma_t^3 \frac{df^3}{d(\log \sigma_0)} \sigma_0^3 \log \left[\frac{N^1 f^1(\tilde{\sigma}_0^1) \sigma_t^1 + N^2 f^2(\tilde{\sigma}_0^2) \sigma_t^2 + N^4 f^2(\tilde{\sigma}_0^4) \sigma_t^2}{(N^2 + N^4) f^2(\sigma_0^{2+4}) \sigma_t^2 + N^3 f^3(\sigma_0^3) \sigma_t^3} \right] .$$

The following features of the mixing error were found after analyzing Eqs. (R-7).

Feature 1: If the common isotope has a constant shielding factor value as a function of σ_0 , no mixing error occurs.

Reasons:

- $E_2 = 0$ because $\frac{df^2}{d(\log \sigma_0)} = 0$.

- $E_1 = E_3 = 0$ because $\tilde{\sigma}_0^1 = \sigma_0^1$ and

$$\tilde{\sigma}_0^3 = \sigma_0^3 \text{ from Eqs. (R-1) and (R-2).}$$

On the other hand, if the gradient of the common isotope shielding-factor is large, the mixing error may be large.

Feature 2: The dominant contributor in the total mixing error is the error caused by the common isotope, E_2 ; that is, $E_2 \gg E_1$ or E_3 .

Reason: The $\log$ argument of E_2 has the factor of $(1 + \text{constant})$, whereas the others have unity.

Feature 3: The maximum error usually is expected to occur when the common isotope is divided approximately evenly between two materials.

$$\text{Reason: } E_2 = A(N^2 \log[B(1 + \frac{N^4}{N^2})] + N^4[B'(1 + \frac{N^2}{N^4})]) .$$

These features were confirmed by numerical experiments using the actual cross section data from LIB-IV. Tables R-I and R-II show characteristics of mixing error resulting from a common isotope that were illustrated by the numerical experiments. The mixed isotopes are ^{238}U , ^{235}U , and ^{16}O . In Example 1,

the separated isotope is ^{238}U , which has a very steep capture shielding factor in group 7 (resonance region: 275 to 3 eV). The results demonstrate that the main contribution of mixing error comes from the common isotope ^{238}U (Feature 2), and the largest error occurs when the common isotope is divided evenly (Feature 3).

TABLE R-I
TYPICAL MIXING ERROR IN THE CAPTURE REACTION
RESULTING FROM ^{238}U AS A COMMON ISOTOPE^a

(7th Group in 9-Group LIB-IV data)

		Exact Solution	Division Ratio of ^{238}U		
			90/10	70/30	50/50
σ_0	^{238}U	8.31	11.8/192.0	18.9/58.9	31.4/31.4
	^{235}U	151.5	169.3	179.9	184.4
	^{16}O	16.0	17.8	18.8	19.3
f	^{238}U	0.0568	0.0638/0.219	0.0806/0.121	0.0986/0.0986
	^{235}U	0.825	0.832	0.836	0.838
	^{16}O	1.002	1.002	1.002	1.002
Σ_{eff}		0.235	0.285	0.314	0.327
$\Delta\sigma_{\text{eff}} (\%)$	^{238}U	Reference	+40%	+63%	+74%
	^{235}U	Reference	+0.8%	+1.3%	+1.6%
	^{16}O	Reference	0.0%	0.0%	0.0%

^aNumber Density of each Isotope

^{238}U : 0.1

^{235}U : 0.01

^{16}O : 0.1

System Temperature = 300 K

TABLE R-II

TYPICAL MIXING ERROR IN THE CAPTURE REACTION
RESULTING FROM ^{16}O AS A COMMON ISOTOPE^a

(7th Group in 9-Group LIB-IV data)

		Exact Solution	Division Ratio of ^{16}O		
			90/10	70/30	50/50
σ_0	^{238}U	8.31	8.31	8.31	8.31
	^{235}U	151.5	151.5	151.5	151.5
	^{16}O	16.0	18.2/193.4	24.4/62.0	35.7/35/7
f	^{238}U	0.0568	0.0568	0.0568	0.0568
	^{235}U	0.825	0.825	0.825	0.825
	^{16}O	1.002	1.002/1.002	1.002/1.002	1.002/1.002
Σ_{eff}		0.235	0.235	0.235	0.235
$\Delta\sigma_{\text{eff}} (\%)$	^{238}U	Reference	0.0%	0.0%	0.0%
	^{235}U	Reference	0.0%	0.0%	0.0%
	^{16}O	Reference	0.0%	0.0%	0.0%

^aNumber Density of each Isotope ^{238}U : 0.1 ^{235}U : 0.01 ^{16}O : 0.1

System Temperature = 300 K

In Example 2, the divided isotope is ^{16}O that has a very flat capture shielding factor in group 7. The results show that there is no effect in the effective cross sections when the common isotope has a flat shielding-factor table (Feature 1).

II. ALTERNATIVE METHOD TO AVOID MIXING ERROR

As shown in the previous section, the mixing error caused by common isotopes has the following characteristics.

- (1) The error becomes serious only when the common isotopes have a steep gradient as a function of the background cross section, σ_0 . This means there is no significant error made in cases where UO_2 is assigned to the fertile material and PuO_2 is assigned to the fissile material because the only common isotope is oxygen and that has a very flat shielding-factor table. This type of material assignment normally is made for SIMMER-II analysis of homogeneous core reactors.
- (2) In the case of an LMFBR heterogeneous core, almost every fissionable isotope becomes a common isotope between fissile and fertile materials, but the dominant isotope generating mixing error is usually only ^{238}U (and perhaps ^{239}Pu).

From these facts, a practical treatment to avoid producing mixing error can be derived. This idea is to separate the dominant isotopes producing the mixing error from the "material" used in SIMMER-II. From the point of view of SIMMER-II, "materials" are composed of "isotopes", as usual. This arrangement is changed only in that some of the "isotopes" are materials in the sense we have used the term.

Table R-III shows an example of the "Dominant-isotope-separation" treatment that typically might be used in a heterogeneous core representation. The total number of isotopes is 14, and the number of fluid-dynamics materials is 5. If we separate ^{238}U and ^{239}Pu as extra materials for neutronics calculation, the efficiency will decrease only by 14%, but the accuracy will improve dramatically over what could be expected with only five materials. Table R-IV shows an example of the accuracy improvement obtainable using this separation method. The error of the five-common-isotopes calculation is 25% in the case of a capture reaction, whereas the error of the ^{238}U -and ^{239}Pu -separated case is only 0.2%.

One of the advantages for this separation method is that a user can trade off efficiency and accuracy according to his objective. For example, if he needs a rigorous solution, he only has to separate all five isotopes as extra materials.

TABLE R-III

ILLUSTRATION OF THE "DOMINANT ISOTOPE SEPARATION TREATMENT"

Case 1. Five neutronics materials containing common isotopes.

Fertile: Mat-A
 Fissile: Mat-B
 Steel: Mat-C
 Control: Mat-D
 Coolant: Mat-E

The materials are as follows.

Mat-A is a mixture of ^{239}Pu , ^{240}Pu , ^{235}U , ^{238}U , and ^{16}O
 Mat-B is a mixture of ^{239}Pu , ^{240}Pu , ^{241}Pu , ^{235}U , ^{238}U , and ^{16}O
 Mat-C is a mixture of Fe, Cr, and Ni.
 Mat-D is a mixture of ^{10}B , ^{11}B , and C.
 Mat-E is a mixture of Na.

Case 2 (Alternative to Case 1). Seven neutronics materials corresponding to the above case but without dominant common isotopes.

Fertile: Mat-A + Mat-F + Mat-G
 Fissile: Mat-B + Mat-F + Mat-G
 Steel: Mat-C
 Control: Mat-D
 Coolant: Mat-E

The materials are as follows.

Mat-A is a mixture of ^{240}Pu , ^{235}U , ^{16}O
 Mat-B is a mixture of ^{240}Pu , ^{241}Pu , ^{235}U , and ^{16}O
 Mat-C is a mixture of Fe, Cr and Ni.
 Mat-D is a mixture of ^{10}B , ^{11}B , and C.
 Mat-E is a mixture of Na.
 Mat-F is a mixture of ^{239}Pu .
 Mat-G is a mixture of ^{238}U .

TABLE R-IV

AN EXAMPLE OF THE ACCURACY IMPROVEMENT POSSIBLE FOR THE OF MIXING METHOD
USING THE COMMON ISOTOPE SEPARATION TREATMENT

(Example based on CRBR heterogeneous core, LIB-IV based 9-group calculation, and 7th group results.)

	<u>Exact Solution</u>	<u>Five Common Isotopes Case</u>	<u>²³⁸U only Separation Case</u>	<u>²³⁸U and ²³⁹Pu Separation Case</u>
$\Sigma_{\text{eff,cap}}$ (error,%)	5.954 (reference)	7.413 (+24.5%)	6.148 (+3.3%)	5.964 (+0.2%)
$\Sigma_{\text{eff,fis}}$ (error,%)	3.110 (reference)	3.300 (+6.1%)	3.272 (+5.2%)	3.110 (0.0%)
$\Sigma_{\text{eff,el,g}\rightarrow\text{g}}$ (error,%)	50.00 (reference)	53.05 (+6.1%)	50.10 (+0.2%)	50.01 (+0.02%)
$\Sigma_{\text{eff,el,g}\rightarrow\text{g}+1}$ (error,%)	0.05443 (reference)	0.05454 (+0.2%)	0.05444 (+0.01%)	0.05443 (0.0%)

APPENDIX S

SYSTEM-DEPENDENT FUNCTIONS AND SUBROUTINES

The system-dependent subroutines will be divided into the following categories.

- General Routines
- Graphic Routines
- LTSS/CTSS Routines
- CDC 7600 Scope Routines
- CRAY COS Routines
- IBM/DEC/FACOM/AMDAHL Routines
- VAX Routines

The routines listed in sections III, IV, V, VI, and VII may be furnished by the FORTRAN libraries or system libraries. If this is not so, the user must supply them.

I. GENERAL ROUTINES

These are routines that the user must supply. Some of these routines have been supplied for the different systems when the procedures are known. The routines used for different options need to be supplied only if the user uses that option.

1. SECOND (T) A subroutine to return in the variable T the elapsed CPU time in seconds since the job started.
2. GETQ (IARG,IOUT) A subroutine to return in IOUT the system parameters requested by IARG. (IARG is a hollerith argument.) Only IARG = 4HKTLM, the job time limit, is used in SIMMER. IOUT for KTLM is the job time limit in seconds.
3. DATE (D) A subroutine to return in the variable D the current calender date as an eight-character literal.

4. CLOCK1 (T) A subroutine to return in the variable T the current time of day as an eight-character literal.
5. LOCF (A) A function to return the address of the first byte of A.
6. DEFILE (N,M,L,K,I) A subroutine to open (if the file exists) or create (if the file does not exist) a direct access file.

Arguments:

Input: N - the data set reference number.
M - the maximum number of records in the data set.
L - the maximum size of each record in bytes, characters, or words (See K.)
K - one of the literals
 1HL (L is bytes)
 1HE (L is characters)
 1HU (L is words).
I - the associate variable.

Output: M = -M if an error occurred when opening or creating the file.

7. SHIFT (I,J) A function to shift the argument I by J bits.
(a) J < 0 I is shifted right J bits. Any bits shifted out the right end are lost. Zero is shifted in the left end.
(b) J = 0 I is returned without any shifting.
(c) J > 0 I is shifted left J bits. Any bits shifted out the left end are shifted in the right end.
This is a left ring shift.

8. MEMEXP (IA,N,I) Memory expansion routine used when the VDM option is specified. This subroutine dynamically expands the amount of memory assigned to the job by N single words. I is returned and indicates the first word of the allocated storage space when referenced to the arbitrary single word array IA; that is, IA(I),....

IA(I+N-1) is the allocated storage space. If I = 0 is returned, the memory expansion failed.

Remark: If the memory expansion is required for N double words referenced to a double-word array A, the routine must be called by CALL MEMEXP (A, 2*N, I) and the returned index I (assumed to indicate a double-word boundary) must be converted into $I' = (I + 1)/2$; then A(I'),...,A(I' + N - 1) is the allocated space.

9. IRANRE (RQT, ILCM, LLEN, NSCT, IUN)
IRANWE (RQT, ILCM, LLEN, NSCT, IUN)

Routines for randomly reading or writing a disk file IUN.

RQT = A 20-word array for use by the routines. RQT is set up by FORQTS routine.

ILCM = LCM pointer giving the first word of LCM to be transferred.

LLEN = The number of words to be transferred.

NSCT = A relative pointer locating the first disk sector. A disk sector contains 512 words.

IUN = Integer logical unit number $1 \leq IUN \leq 99$.

10. IDONEQ (IUN) A function to test whether random I/O unit IUN is finished with the previous read or write.
IDONEQ = 0, unit is not finished.
IDONEQ $\neq$ 0, unit is finished.
11. FORQTS (fname, RQT) Create a file called "fname" and initialize the RQT request table for file "fname." RQT is a 20-word array for use by the system.
12. TICHEK (i,j,k,l,m) Timing routine used when the STAT option is specified for the SIMMER preprocessor.
i = hollerith output of the form

bb = blank
 xx = percentage of time for CPU usage
 yy = percentage of time for I/O
 zz = percentage of time for system calls
 j = number of μ s since last call to TICHEK
 k = total number of CPU μ s since start of execution
 l = total number of I/O μ s since start of execution
 m = total number of system μ s since start of execution

13. ECRD (SM,LM,N,IER) Routines used when the NOLCM option is specified.
 ECWR (SM,LM,N,IER) ECRD/ECWR moves N words of real data from/to the AA1
 array location LM in blank block to/from the SM array.

14. IECRD (ISM,ILCM,N,IEREC)
 IECWR (ISM,ILCM,N,IEREC)
 Routines used when the NOLCM option is specified.
 IECRD/IECWR moves N words of integer data from/to the
 IAA1 array location ILCM in the blank common block to/
 from the ISM array.

15. CVMGN (X_1, X_2, X_3) Conditional vector merge function. This is intended to
 be an in-line function supplied by the compiler on a
 vector machine. X_1 is returned if $X_3 \neq 0$; otherwise
 X_2 is returned. Used only if NEUPREP is defined.

16. GATHER (m,a,b,index) Used to gather data into contiguous storage to enable
 loop vectorization. Performs the following FORTRAN
 function.
 $A(I) = B(\text{INDEX}(I))$, $I = 1, \dots, M$, where
 M = number of elements in each vector,
 A = output vector,
 B = source vector, and
 index = vector of indexes giving the desired source
 vector elements.

17. SCATTER (m,a,index,b)

Used to scatter data into the desired array locations from contiguous storage to enable loop vectorization. Performs the following FORTRAN function.

A INDEX (I) = B(I), I = 1, ..., M , where

M = number of elements in each vector,

A = output vector,

B = source vector, and

index = vector of indexes giving the desired output vector elements.

18. ISAMAX (m,sx,incx) A function returning the first index i such that

$|X_i| = \max |X_j|$ for $j = 1, \dots, m$, where

X_j = the element of a real vector,

m = number of elements to be processed in the vector to be searched,

sx = real vector to be searched, and

incx = skip distance between elements of sx.

For contiguous elements, incx = 1.

II. GRAPHIC ROUTINES

These routines are used by the SIMMER-II plotting features. (The NOTPLT option is not specified.) The list of routines are those used by the LTSS/CTSS systems at Los Alamos. The routines will be divided into two groups: initialization and closing of the graphics system and plotting routines. The allowable character set used by these routines is as follows.

<u>Character</u>	<u>Decimal</u>	<u>Character</u>	<u>Decimal</u>
0	0	R	41
1	1	(plotting	
2	2	.dot)	42
3	3	\$	43
4	4	*	44
5	5	α	45
6	6	~	46
7	7	d	47
8	8	(blank)	48
9	9	/	49
@	10	S	50
=	11	T	51

<u>Character</u>	<u>Decimal</u>	<u>Character</u>	<u>Decimal</u>
"	12	U	52
,	13	V	53
fδ	14	w	54
α	15	x	55
+	16	y	56
A	17	Z	57
B	18	^o (degree)	58
C	19	,	59
D	20	(	60
E	21	/	61
F	22	Σ	62
G	23	[]	63
H	24		
I	25		
π	26		
.(period)	27		
)	28		
β	29		
+	30		
μ	31		
-	32		
J	33		
K	34		
L	35		
M	36		
N	37		
O	38		
P	39		
Q	40		

A. Initialization and Closing Routines

- (a) GPLOT (ICL, LABEL, 60) Initialize the LTSS/CTSS graphics system.
 ICL - classification of the graphics file.
 1HU - unclassified
 1HC - confidential
 1HS - secret

NOTE: This graphics system uses a 1024 x 1024 grid. The point (0,0) is the upper right-hand corner. The point (1023, 1023) is the lower left-hand corner. All routines sending commands to the graphics system send the

coordinates as integers, that is, (IX,IY),
 $0 \leq IX \leq 1023$, and $0 \leq IY \leq 1023$.

(b) LIB4020

Initializes the LTSS/CTSS graphics system to allow a formatted FORTRAN write statement to the unit number specified by GRPHLUN to be sent to the graphics file, that is, allows the mixing to text and plots.

(c) GRPHLUN (LUN)

Initializes LUN as the graphics file unit number. ($1 \leq LUN \leq 99$).

(d) GPOLAY

Initializes the graphics system for use with overlaid codes. GPOLAY must be called in the first overlay.

(e) GDONE

Terminate all plotting and empty all graphics buffers.

B. Plotting Routines

(a) ADV(N)

Advance Film

N - Number of frames to advance.

(b) WLCH(IX,IY,NC,BCD,ISIZE)

WLCV(IX,IY,NC,BCD,ISIZE)

The NC characters stored starting at BCD will be written either horizontally or vertically with the upper left corner of the first character at (IX,IY) for horizontal characters and the lower left corner at (IX,IY) for vertical characters. Character and line spacing is automatic in either direction. Each line is spaced down by $7(ISIZE+1)+1$ plotting positions as pertains to corresponding points in the letter matrices. The total space used by each character by size including boundary area is as follows.

PLOTTING POSITIONS OCCUPIED

<u>ISIZE</u>	<u>HORIZONTAL</u>	<u>VERTICAL</u>	<u>CHARACTERS/LINE</u>
1	12	15	85
2	18	22	57
3	24	19	42
4	30	36	34
5	36	43	28
6	42	50	24

(c) DRV(IX1,IY1,IX2,IY2)

A straight-line vector will be drawn from (IX1,IY1) to (IX2,IY2).
Example: CALL DRV(512,512,1023,0) will cause a straight line to be drawn from the center (512,512) to the upper right-hand corner (1023,0).

(d) PLT(IX,IY,ICHAR)

This subroutine plots the character specified by ICHAR at the location (IX,IY). Table R-III is a list of the 64 available characters with their integer codes.

Example: CALL PLT(512,512,43) will cause a dollar sign (decimal integer code = 43) to be plotted at the center of the grid, location (512,512).

(e) GXA(IX1,IX2,IY) Generate an x-axis. A horizontal axis will be drawn from (IX1,IY) to (IX2,IY). Example: CALL GXA(0,512,0) or CALL GXA(512,0,0) will cause a horizontal axis to be drawn from (0,0) to the point (512,0) or halfway across the top from the top left corner to the center.

(f) GYA(IY1,IY2,IX) Generate a y-axis. A vertical axis will be drawn from (IY1,IX) to (IY2,IX). Example: CALL GYA (0,512,512) or CALL GYA(512,0,512) will produce a vertical axis from the center of the grid (512,512) to the top of the grid (512,0).

(g) TSP(IX,IY,NC,BCD)

A message consisting of NC consecutively stored BCD characters will be "typed" out starting at coordinate (IX,IY). Each suc-

ceeding character is typed 8 plotting positions to the right of the preceding character until the line is full at, which point the next character will be spaced down 1 line (16 plotting positions) and at the leftmost horizontal position (same as a carriage return).

IF IX or IY are out of bounds (< 0 or > 1023), their values will be set to the corresponding maximum or minimum value, and typing will proceed at that point.

(k) COLOR(F)

Select color filter(s) for film-writing devices.

F - filter selection.

<u>Color</u>	<u>Value of F</u>	<u>Filters used</u>
White	0.0	No filter
Red	1.0	Magenta and yellow
Yellow	1.5	Yellow
Green	2.0	Cyan and yellow
Cyan (blue green)	2.5	Cyan
Blue	3.0	Cyan and magenta
Magenta	3.5	Magenta

III. LTSS/CTSS ROUTINES

The LTSS/CTSS systems as used with SIMMER-II are from the systems used at Los Alamos. The LTSS system is on the CDC-7600 computer, and the CTSS system is on the CRAY-1S computer. If the user has a LTSS or CTSS system, the system library may have the routines. If the system library does not have the routines, the user will have to supply them. The options to extract code for the LTSS system are FTNLTS, FTN42, OVLAY, LCM, and TENB. The options to extract code for the CTSS system are FTNLTS, FTN42, CFT, NOLCM, LMEM, and EIGHTB. Please note that the CRAY option is not used for the CTSS option. The LTSS/CTSS routines are as follows.

(1) OVLCMIP(LTAB,N)

This routine is used only with LTSS. It loads N overlays listed in LTAB from LCM if possible.

(2) IOTENAB	Enable the I/O timer.
(3) IOTIME	Print the I/O times enabled by IOTENAB.
(4) EXIT	Terminate execution after emptying all I/O buffers.
(5) GETFL (LSCM,LLCM)	Get the length of SCM, LSCM, and the length of LCM, LLCM, being used by SIMMER-II. On CTSS, only LLCM is returned, giving the length of all memory used.
(6) SETFLG (LUN,I)	Set I/O unit LUN flags to I. If I = 0, do not create file LUN. If the buffers overflow, terminate the job. If I = 1, create the file LUN only if one-half of the buffer overflows.
(7) OVERLAY (IFN,IP,IS,IR)	A subroutine to load the requested overlay into memory. IFN - The file containing the overlay. IP - The primary overlay number. IS - The secondary overlay number. IR - The recall flag.
(8) CLOSE (LUN)	Close out unit LUN.
(9) FAMSIZ (LUN,LEN)	Change the family file size to LEN on unit LUN.
(10) SSWTCH (ISNO,ISFLG)	Test if sense switch ISNO is on or off ISFLG = 0, switch is off; ISFLG = 1, switch is on.
(11) MOVLEV (LC,SC,NWDS)	Move NWDS words from LCM location LC to SCM location SC.
MOVLEV (SC,LC,NWDS)	Move NWDS words from SCM location SC to LCM location LC.

(12) GETJTL (T)

Get the job time limit from the system.

(13) QADJUST(NAME, IDEST, IFLS, IFLL, IERROR)

Routine to change the name of the existing dropfile and/or modify the field lengths if desired.

NAME - Name of the New Dropfile

- If 0, a new unique name will be created.

- If -1, the old name will be used. IDEST is ignored, and if the old dropfile will not hold the new image, the old one will be destroyed and recreated at the new larger size.

IDEST - If 1, the old dropfile will be destroyed.

If 0, the old dropfile will be preserved unless NAME = -1.

IFLS - New SCM files length.

- If 0, the old value will be used

IFLL - New LCM field length.

- IF 0, the old value will be used

IERROR - Error Flag

- 0 - Normal termination

- 1 - Could not get PP name and FLL

- 2 - Dropfile create failed

- 3 - Could not adjust FLS

- 4 - Could not adjust FLL

- 5 - Could not destroy old dropfile

(14) RDISK (NUNIT, BUF, NWDS, IDS) Random read (RDISK)/write (WDISK) NWDS words
WDISK (NUNIT, BUF, NWDS, IDS) from/to unit NUNIT to/from the buffer BUF.

(15) EOF (NUNIT)

Logical function to test an end-of-file condition on unit NUNIT.

(16) UNIT (NUNIT)	Function to test unit idle, end-of-file condition, or error condition.
(17) QASSIGN (NUNIT, NAME,
BUF,W)	Assign unit NUNIT to a file named NAME using NW words of the array BUF as I/O buffer. If BUF is 0 and NW is 0, buffers will not be used.
(18) SKIPR (NUNIT, NREC)	Skip NREC records on unit NUNIT. If NREC < 0, skip backwards. If NREC > 0, skip forward.
(19) CLOCK1 (T)	This subroutine returns in T the current time of day as an eight-character literal.
(20) DATEH(D)	Subroutine to give the current calendar date D as an eight-character BCD word.

IV. CDC-7600 SCOPE ROUTINES

The system library on the SCOPE system may have the following routines. If the system library does not have the routines, the user will have to supply them. The option to extract code for the SCOPE system are FTN42, SCOPE, OVLAY, LCM, and TENB.

(1) EXIT	Terminate execution after emptying all I/O buffers.
(2) OVERLAY (IFN,IP,IS,IR)	A subroutine to load the requested overlay into SCM ready for execution.

IFN = The file name containing the overlay
IP = The primary overlay number
IS = The secondary overlay number
IR = The recall flag |
| (3) Q9RET (LUN) | Close out I/O unit LUN. |
| (4) SSWTCH (ISNQ,ISFLG) | Test if sense switch ISNQ is on or off. |
| (5) READEC(SC,LC,N)
WRITEC(SC,LC,N) | Read/write N words from/to LCM array LC to/from SCM array SC. |
| (6) TIM2GO | Get from the system the task time limit (time remaining for the task to execute). |

- (7) DATE (D) This subroutine returns in D the current calendar dates as an eight-character literal.
- (8) CLOCK (T) This subroutine returns in T the current time of day as an eight-character literal.
- (9) OPENMS (LUN,IX,L,T) Open a random file
 LUN = limit number
 IX = Name of the master index array
 L = Length of master index array
 T = Type of index.
- (10) READMS (LUN,LC,N,IR) Read/write from/to the random file LUN N words
 WRITMS (LUN,LC,N,IR) to/from the LCM array LC.

NOTE: OPENMS, READMS and WRITEMS are FTN library subroutines.
 Checked the FTN manual for a complete description.

V. CRAY COS ROUTINES

The system library on the COS system may have the following routines. If the system library does not have the routines, the user will have to supply them. The options to extract code for the COS system are CRAY, CFT, NOLOM, LMEM, and EIGHTB

- (1) OPEN (LUN) Open File LUN.
- (2) DISPOSE (LUN) Close out file LUN.
- (3) GETTL (T) Get the task time limit from the system.
- (4) DATE (D) This subroutine returns in D the current calendar date as an eight-character literal.
- (5) CLOCK (T) This subroutine returns in T the current time of day as an eight-character literal.
- (6) RDR (LUN,LC,N,NDPT) Random read (RDR)/write (RDW) from/to unit LUN, location NDPT, N words to/from buffer LC.

VI. IBM, DEC 10, DEC 11, AMDAHL, and FACOM ROUTINES

The system library for these systems may have the following routines. If the system library does not have the routines, the user will have to supply them. The options to extract code for these systems are IBM, DEC, DBL, NOLCM, LMEM, and FOURB.

- (1) WAIT(I,ID,COND) IBM I/O delay function.
 I = Logical unit number.
 ID = I/O identification.
 COND = I/O completion condition.
- (2) DSDEF (N,M) This subroutine returns in M the space reserved on data set N by the JCL data definition statement in number of records.
- (3) DATE (D) This subroutine returns in D the current calendar date as an eight-character literal.
- (4) CLOCK1 (T) This subroutine returns in T the current time of day as an eight-character literal.

VII. VAX ROUTINES

The system library on the VAX system may have these routines. If the system library does not have these routines, the user will have to supply them. The options to extract code for the VAX system are IBM, DEC, VAX, DBL, NOLCM, LMEM, and FOURB.

- (1) TIME(T) Get the current time of day as eight-character literal.
- (2) IDATE (I,J,K) Returns the month I, day J, and year K as integer.
- (3) SYS\$EXPREG(IPC,IA,IM,IR) Expand visual memory IPC pages of mode IM. IR is the region to be expanded. IA is a word array where the starting and ending page address is returned.
- (4) SYS\$GETJPI A system routine to get a job process information. Used to get the accumulated CPU time in 10-ms ticks.