ANL-92/33

ANL--92/33

DE93 006044

**ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439**

**COMMIX-1AR/P: A THREE-DIMENSIONAL TRANSIENT
SINGLE-PHASE COMPUTER PROGRAM
FOR THERMAL HYDRAULIC ANALYSIS
OF SINGLE AND MULTICOMPONENT SYSTEMS**

Volume 2: User's Guide

by

P. L. Garner, R. N. Blomquist, and E. M. Gelbard

Reactor Analysis Division

September 1992

MASTER

EB
DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

PREFACE

The documentation of the computer program COMMIX-1AR/P has been divided into four volumes, each addressing different aspects of the documentation process and written from different perspectives. A brief summary of the four volumes is as follows:

Volume 1: "Equations and Numerics" (ANL-90/45)

Presents the basic equations, assumptions, finite-difference techniques, and solution methods.

Volume 2: "User's Guide" (ANL-92/33)

A step-by-step description of how to use the program, including an input guide and a sample problem.

Volume 3: "Programmer's Guide" (ANL-92/34)

A description of each subroutine and variable, showing linkage among these and their relation to the equations and variables presented in Volume 1.

Volume 4: "Verification and Validation"

A collection of representative problems run on this computer program and a comparison of the results obtained to those from theory, experiments, and other computer programs.

The normal user will primarily use the information in Volumes 1, 2, and 4 and will be using a controlled configuration of the program. Volume 3 is designed as a consolidated source of reference material for those involved in the program development effort.

TABLE OF CONTENTS

	Page
NOMENCLATURE	xiv
ABSTRACT	xix
EXECUTIVE SUMMARY	xix
1. INTRODUCTION	1-1
2. INPUT FILES AND DATA FORMATS	2-1
3. GEOMETRY	3-1
3.1 Grid Definition	3-1
3.2 Grid Orientation	3-4
3.3 Fluid Assignment	3-5
3.4 Boundary Surface Definition	3-7
3.5 Volume and Surface Porosities	3-16
4. BOUNDARY AND INITIAL CONDITIONS	4-1
4.1 Boundary Conditions	4-1
4.1.1 Velocity	4-2
4.1.2 Pressure	4-9
4.1.3 Energy	4-12
4.1.4 Transient Functions	4-19
4.1.5 User Input	4-24
4.2 Initial Conditions	4-31
4.2.1 Initialization Options	4-31
4.2.2 User Input	4-33
5. THERMAL STRUCTURES	5-1
5.1 Review of Model Features	5-1
5.2 Thermal Structure Prototype Input Block	5-6
5.2.1 NAMELIST T	5-9
5.2.2 NAMELIST F	5-14
5.2.3 NAMELIST M	5-15
5.3 Thermal Structure Location Input Block	5-19
5.4 Additional User Input	5-28
6. FORCE STRUCTURES	6-1
6.1 Review of Model Features	6-1
6.2 User Input	6-9

TABLE OF CONTENTS (Cont'd)

	Page
7. PUMPS	7-1
7.1 Selecion of Model	7-1
7.2 Homologous Model	7-3
7.3 Specified Speed Model	7-14
7.4 Specified Pressure Increase Model	7-18
7.5 Pump Location Input Block	7-19
8. TURBULENCE	8-1
8.1 Selection of Model	8-1
8.2 User-Specified Models	8-3
8.3 Two-Equation k- ϵ Model	8-5
8.4 One-Equation k Model	8-10
8.5 Zero-Equation Mixing-Length Model	8-12
9. PROPERTIES AND HEAT TRANSFER COEFFICIENTS	9-1
9.1. Fluid Properties	9-1
9.2. Structure Properties	9-6
9.3. Heat Transfer Coefficients	9-10
10. HEXAGONAL FUEL ASSEMBLY	10-1
10.1 Geometry	10-1
10.2 Forces from Fuel Pins and Wire Wraps	10-8
10.3 Heat Generation	10-10
10.4 Other Considerations	10-14
11. MASS REBALANCING	11-1
11.1 User-Specified Rebalancing	11-1
11.2 Plane-by-Plane Rebalancing	11-8
12. AUXILIARY INPUT	12-1
12.1 Starting, Stopping, and Restarting	12-1
12.2 Selection of Solution Methods	12-6
12.3 Time Step Size Selection	12-11
12.4 Iteration Controls and Parameters	12-15
12.5 Steady State	12-23
12.6 Output Controls	12-25
ACKNOWLEDGMENTS	Ref-1
REFERENCES	Ref-2

TABLE OF CONTENTS (Cont'd)

	Page
APPENDICES	
A. INPUT GUIDE	A-1
B. DETAILED FLUID PROPERTY EXAMPLES	B-1
B.1 Liquid Sodium	B-1
B.2 Liquid Normal Water	B-4
B.3 Liquid Heavy Water	B-10
B.4 Helium Gas	B-12
C. INSTALLATION AND EXECUTION INSTRUCTIONS	C-1
C.1 Installation	C-1
C.1.1 Background	C-2
C.1.2 Specific Instructions	C-8
C.1.2.1 Cray	C-9
C.1.2.2 IBM	C-25
C.2 Execution	C-42
C.2.1 Background	C-42
C.2.2 Specific Instructions	C-45
C.2.2.1 Cray	C-45
C.2.2.2 IBM	C-47
D. GRAPHICS POSTPROCESSORS	D-1
D.1 Introduction	D-1
D.2 Program VECT	D-3
D.3 Program TEMP	D-12
D.4 Program TDSC	D-21
D.5 Program GRPH	D-28
E. SAMPLE PROBLEM	E-1
E.1 Description	E-1
E.2 Card-Image Input File	E-3
E.2.1 Geometry	E-10
E.2.2 Boundary and Initial Conditions	E-18
E.2.3 Thermal Structures	E-20
E.2.4 Force Structures	E-25
E.2.5 Pump	E-27
E.2.6 Turbulence	E-28
E.2.7 Properties and Heat Transfer Coefficients	E-28
E.2.8 Auxiliary	E-29
E.3 Printable Output File	E-31
E.3.1 Case Summary	E-31
E.3.2 Short Edit	E-36
E.3.3 Long Edit	E-41
E.4 Graphics Output	E-89
E.5 Continuation	E-98

LIST OF FIGURES

	Page
3.1 Partitioning of Flow Domain into Computational Cells for Two Coordinate Systems: (a) Cartesian and (b) Cylindrical	3-3
3.2 Example Geometry Showing (a) Boundary Surface Numbering and (b) Partitioning into Boundary Surface Elements	3-9
3.3 Example Geometry Having an Irregular Boundary Surface: (a) Surface Numbering and (b) Partitioning into Surface Elements	3-10
3.4 Example of Noncoincident Surfaces Having the Same Boundary Surface Number (2)	3-12
3.5 Examples of Coincident Surfaces Having Different Boundary Surface Numbers (6 vs. 5 and 8 vs. 3)	3-12
3.6 Examples Showing Partitioning of Complex Geometries	3-14
3.7 Alteration of Volume and Surface Porosities by (a) Irregular Geometry and (b) Internal Solid	3-19
4.1 Cells Near Outlet Boundary Surfaces	4-5
4.2 Example Placement of an Expansion Cell	4-8
4.3 Recommended Boundary Surface Arrangements for Pressure Boundary Conditions	4-11
4.4 Finite-Thickness Wall Boundary Surface	4-18
4.5 Examples of Transient Functions	4-23
5.1 Definition of a Cylindrical Thermal Structure Element Relative to the Flow Domain	5-2
5.2 Outer and Inner Surfaces of Several Thermal Structure Elements in the Fluid Grid Structure	5-3
5.3 Four Quarter-Cylindrical Thermal Structure Elements, Each Interacting with One Fluid Cell	5-3
5.4 Multiple Thermal Structures Interacting with a Single Fluid Cell	5-5
5.5 Material Regions and Gaps for a Typical Thermal Structure Element	5-5
5.6 Use of Nonunity RODFR Values in Assigning a Thermal Structure to a Cell: (a) Multiple (RODFR=10.0) and (b) Fractional (RODFR=0.25)	5-13
5.7 Use of a Negative RODFR Value ($=-2/(a*b)$) in Assigning a Uniform Distribution of a Thermal Structure in a Nonuniform Grid	5-13
5.8 Use of a Negative RODFR Value ($=-1/(\pi*(r_3^2-r_2^2))$) in Assigning a Cylindrical Shell Thermal Structure in a Nonuniform Azimuthal Grid	5-13
5.9 Projections of Grid Geometry Showing Placements of Slab- and Cylinder-Type Thermal Structures	5-27
6.1 Illustration of Staggered Grid Concept, Showing Difference Between Main and Momentum Control Volumes	6-7

LIST OF FIGURES (Cont'd)

	Page
6.2 Flow Through a Sudden Enlargement: (a) Basic Geometry, (b) Placement in Grid with Reference Velocity u_1 , and (c) Placement in Grid with Reference Velocity u_2	6-8
6.3 Example of Force Structure Location Relative to the Two Types of Control Volumes: (a) Momentum and (b) Main	6-17
7.1 Default Characteristic Head (WH) and Torque (WT) Functions Used in the Homologous Pump Model (IFPUMP=3). (Symbols are data for $N_s=35$ (or 1800 -- see text) from Ref. 15 and lines are fits to these data from Ref. 16.)	7-8
10.1 Quarter-Pin Partitioning of a 19-Pin Fuel Assembly in a Hexagonal Duct	10-5
10.2 Full-Pin Partitioning of a 19-Pin Fuel Assembly in a Hexagonal Duct	10-5
11.1 Example of User-Specified Mass Rebalancing Regions	11-2
E.1 Elevation View of Geometry Grid for Sample Problem for Planes J=3 (left) and J=1 (right)	E-12
E.2 Elevation View of Geometry Grid for Sample Problem for Plane I=2	E-13
E.3 Plan View of Geometry Grid for Sample Problem for Plane K=9	E-14
E.4 Steady-State Velocity Distribution for Sample Problem for Planes J=3 (left) and J=1 (right)	E-90
E.5 Steady-State Velocity Distribution for Sample Problem for Plane K=10	E-91
E.6 Steady-State Temperature Distribution (Isotherm Spacing of 10°C) for Sample Problem for Planes J=3 (left) and J=1 (right)	E-93
E.7 Steady-State Temperature Distribution for Sample Problem for Plane K=9	E-94
E.8 Velocity Histories During Approach to Steady State for Sample Problem	E-96
E.9 Temperature Histories During Approach to Steady State for Sample Problem	E-97

LIST OF TABLES

	Page
2.1 Definition of ISTATE	2-1
2.2 Ordering of Blocks in the Card-Image Input File	2-2
3.1 Grid Definition Variables in NAMELIST GEOM	3-2
3.2 Grid Orientation Variables in NAMELIST DATA	3-5
3.3 Fluid Use Variable in NAMELIST GEOM	3-5
3.4 Fluid Cell Location Input Block Record Structure	3-6
3.5 Unit Surface-Normal Vectors for Figure 3.2	3-8
3.6 Unit Surface-Normal Vectors for Figure 3.3	3-8
3.7 Boundary Surface Definition Variables in NAMELIST GEOM	3-15
3.8 Boundary Surface Definition Variables in NAMELIST DATA	3-15
3.9 Boundary Surface Identification Input Block Record Structure	3-17
3.10 Geometry Variables in Internal Cell Initialization Input Block	3-22
4.1 Types of Velocity Boundary Conditions	4-3
4.2 Types of Pressure Boundary Conditions	4-10
4.3 Types of Energy Boundary Conditions	4-14
4.4 Transient Function Definition Variables in NAMELIST DATA	4-20
4.5 Boundary Condition Variables in NAMELIST DATA	4-25
4.6 Additional Energy Boundary Condition Variables in NAMELIST DATA	4-27
4.7 Boundary Condition Variables in NAMELIST INPUTQ	4-28
4.8 Initial Condition Variables in NAMELIST DATA	4-34
4.9 Initial Condition Variables in NAMELIST GEOM	4-35
4.10 Initial Condition Variables in Internal Cell Initialization Input Block	4-36
4.11 Boundary Value Initialization Input Block Record Structure	4-38
5.1 Thermal Structure Prototype Input Block Structure	5-8
5.2 Thermal Structure Variables in NAMELIST T which Begins the Definition of a Prototype	5-10
5.3 Thermal Structure Variables in NAMELIST F which Defines Heat Transfer to Fluid from the Outside and/or Inside Surface of a Prototype	5-14
5.4 Thermal Structure Variables in NAMELIST M which Defines a Material Region within a Prototype	5-16
5.5 Thermal Structure Location Input Block Record Structure	5-20
5.6 Some of the Acceptable Forms for Thermal Structure Location Input Block Records for Slab-Type Thermal Structure Number 10 Separating Fluid Cells ($2 \leq I \leq 4, J=5, K=8$) from Fluid Cells ($2 \leq I \leq 4, J=5, K=9$)	5-23
5.7 Thermal Structure Variables in NAMELIST GEOM	5-29
5.8 Thermal Structure Variables in NAMELIST DATA	5-30
5.9 Thermal Structure Variables in NAMELIST STRUCT	5-35
6.1 Friction Factor Correlations in Library	6-4
6.2 Force Structure Variables in NAMELIST GEOM	6-10
6.3 Force Structure Variables in NAMELIST DATA	6-11
6.4 Force Structure Location Input Block Record Structure	6-16
7.1 Pump Model Variables in NAMELIST DATA	7-2

LIST OF TABLES (Cont'd)

	Page
7.2 Steady-State Homologous Pump Model (IFPUMP=3 or 4) Variables in NAMELIST PUMP	7-3
7.3 Quadrant Assignments for X (Eq. 7.4)	7-6
7.4 Transient Homologous Pump Model (IFPUMP=3 or 4) Variables in NAMELIST PUMP	7-11
7.5 Specified-Speed Pump Model (IFPUMP=1) Variables in NAMELIST PUMP	7-15
7.6 Specified-Pressure-Increase Pump Model (IFPUMP=2) Variables in NAMELIST PUMP	7-18
7.7 Pump Location Input Block Record Structure	7-21
8.1 Turbulence Model Variables in NAMELIST GEOM	8-2
8.2 User-Specified Turbulence Model (ITURKE=0 or 1) Variables in NAMELIST DATA	8-3
8.3 User-Specified Turbulence Model (ITURKE=1) Variables in Internal Cell Initialization Input Block	8-6
8.4 Two-Equation ("k- ϵ ") Turbulence Model (ITURKE=12) Variables in NAMELIST DATA	8-8
8.5 One-Equation ("k") Turbulence Model (ITURKE=11) Variables in NAMELIST DATA	8-11
8.6 Zero-Equation "Mixing-Length" Turbulence Model (ITURKE=10) Variables in NAMELIST DATA	8-12
9.1 Fluid Property Variables in NAMELIST DATA	9-2
9.2 Functional Dependence for the Detailed Fluid Properties and the Names and Arguments of the Corresponding FORTRAN FUNCTIONS	9-4
9.3 Structure (Solid) Property Variables in NAMELIST DATA	9-7
9.4 Entry Point Description for FUNCTION CPSOL1	9-9
9.5 Entry Point Description for FUNCTION XKSOL1	9-9
9.6 Heat Transfer Correlation Variables in NAMELIST DATA	9-12
9.7 Entry Point Description for FUNCTION HEATCU	9-13
10.1 Hexagonal Fuel Assembly Variables in NAMELIST GEOM	10-2
10.2 Boundary Surface Assignments for Hexagonal Fuel Assembly Geometry Option (IGEOM>0)	10-4
10.3 Other Geometry Variable Values Automatically Set for Hexagonal Fuel Assembly Option (IGEOM>0)	10-6
10.4 Hexagonal Fuel Assembly Variables in NAMELIST DATA	10-9
10.5 Hexagonal Fuel Assembly Variables in NAMELIST INPUTQ	10-12
10.6 Typical Boundary Conditions for Hexagonal Fuel Assemblies	10-14
11.1 User-Specified Rebalancing Variables in NAMELIST GEOM	11-3
11.2 User-Specified Rebalancing Variables in NAMELIST DATA	11-3
11.3 Rebalancing Region Location Input Block Record Structure	11-4
11.4 Portions of Input for User-Specified Rebalancing Example Shown in Figure 11.1	11-7
11.5 Plane-by-Plane Rebalancing Variables in NAMELIST DATA	11-9

LIST OF TABLES (Cont'd)

	Page
12.1 Input Variables Associated with Starting, Stopping, and Restarting a Case in NAMELISTs GEOM and DATA	12-2
12.2 Conditions which Define the "Normal End" of a Computer Run	12-3
12.3 Input Variables Associated with Solution Methods in NAMELISTs GEOM and DATA	12-7
12.4 Examples of Calculational Frequencies Selected by ISETMO and ISETEN	12-10
12.5 Input Variables Associated with Time Step Size Selection in NAMELIST DATA	12-12
12.6 Input Variables Associated with Iteration Control in NAMELIST DATA	12-16
12.7 Input Variables Associated with Case Summary Output Control in NAMELISTs GEOM and STRUCT	12-26
12.8 Input Variables Associated with Debug Output Control in NAMELISTs DATA and PUMP	12-27
12.9 Input Variables Associated with Normal Output Control in NAMELIST DATA	12-29
12.10 Identification Codes (VV) for Boundary Surface Arrays as Used in Input Arrays ISTPR and NTHPR	12-32
12.11 Identification Codes (VV) for Internal Arrays as Used in Input Arrays ISTPR and NTHPR	12-33
C.1 Special System Routines Needed and Locations Assumed in Sample Jobstreams	C-4
C.2 FORTRAN-77 Example of FUNCTION TLEFTS for Use in Interactive Mode	C-6
C.3 Code Sections to Be Installed	C-8
C.4 Cray Batch Job to Compile Code Section	C-11
C.5 Cray Batch Job to Execute COMMIX	C-13
C.6 Cray Batch Job to Execute Graphics Postprocessor Codes VECT or TEMP	C-16
C.7 Cray Batch Job to Execute Graphics Postprocessor Code TDSC	C-19
C.8 Cray Batch Job to Execute Graphics Postprocessor Code GRPH	C-22
C.9 IBM Batch Job to Compile Code Section	C-29
C.10 IBM Cataloged Procedure FORTVCP at ANL for Compilation	C-29
C.11 IBM Batch Job to Execute COMMIX	C-30
C.12 IBM Batch Job to Execute Graphics Postprocessor Codes VECT or TEMP	C-32
C.13 IBM Batch Job to Execute Graphics Postprocessor Code TDSC	C-34
C.14 IBM Batch Job to Execute Graphics Postprocessor Code GRPH	C-36
C.15 IBM Cataloged Procedure FORTVCEG at ANL for Compile, Link Edit, and Execution	C-38
C.16 IBM Cataloged Procedure FORTVCEP at ANL for Compile and Link Edit	C-40
C.17 IBM Cataloged Procedure RUN at ANL for Execution	C-41

LIST OF TABLES (Cont'd)

	Page
D.1 Graphics Postprocessor Programs	D-2
D.2 Input Guide for Program VECT	D-8
D.3 Sample Card-Image Input File for Program VECT	D-11
D.4 Input Guide for Program TEMP	D-17
D.5 Sample Card-Image Input File for Program TEMP	D-20
D.6 Input Guide for Program TDSC	D-25
D.7 Sample Card-Image Input Files for Program TDSC	D-27
D.8 Input Guide for Program GRPH	D-33
D.9 Sample Card-Image Input File for Program GRPH	D-36
E.1 Listing of Card-Image Input File	E-4
E.2 Boundary Surface Summary	E-15
E.3 Listing of Printable Output File	E-48
E.4 Card-Image Input File to Continue a Steady-State Calculation	E-98
E.5 Card-Image Input File to Begin a Transient Calculation from a Steady-State Calculation	E-99
E.6 Card-Image Input File to Continue a Transient Calculation	E-99

Nomenclature →

NOMENCLATURE

A_M	flow area for surface whose unit normal vector points in direction M, where M is x or r, y or θ , or z	K
a_i^ϕ	coefficient multiplying unknown ϕ_i in general form of conservation equation for ϕ	k
b_0^ϕ	constant term in general form of conservation equation for ϕ	L
C	constants in various correlations (e.g. friction, heat transfer, pump)	ℓ
C_p	specific heat capacity	N
D_h	characteristic length (e.g. hydraulic diameter)	N
d_i^ϕ	coefficient multiplying ΔP in equation for velocity ϕ which results from the momentum equation	n
d_w	wire wrap diameter	P
E	kinetic energy	P_e
$F_{L,M}$	fraction of energy leaving radiation surface number L which arrives at radiation surface number M ("view factor" in Sec. 5.4)	P_r
f	friction factor	P_x
$f_{nt}(t)$	transient function number nf	Q
f_μ	turbulent viscosity coefficient function	q
$\vec{g}$	gravitational acceleration vector, having components g_x , g_y , and g_z in the x, y, and z directions, respectively	q^+
H	head developed by pump	R_z
h	enthalpy	R_z
I_p	moment of inertia for pump	r
I (or i)	spatial location cell index corresponding to x or r direction	r_p
$\vec{i}_M$	unit vector in direction M, where M is x or r, y or θ , or z	T
J (or j)	spatial location cell index corresponding to y or θ direction	t
		U
		u

(or k) spatial location cell index corresponding to z direction

turbulent kinetic energy

length or thickness

length scale in zero- and one-equation turbulence models

rotational speed of pump

Nusselt number

unit surface normal vector (points toward fluid), having components n_x or n_r , n_y or n_θ , and n_z in the i , j , and k directions, respectively

pressure

saturation pressure

wire wrap pitch

P_x , P_y fuel pin pitch in x and y directions, respectively

Prandtl number

heat transfer rate (general); also, volumetric flow rate through pump (Sec. 7)

heat transfer rate per unit area ("flux")

heat generation or transfer rate per unit volume

distributed resistance term in the momentum equation for direction M , where M is x or r , y or θ , or z

Reynolds number

i -direction spatial coordinate in cylindrical system ("radius")

radius of fuel pin

temperature

time

heat transfer coefficient

velocity component in x (or r) direction (i.e. u_x or u_r)

u_M	velocity component in direction M, where M is x or r, y or θ , or z (general)	λ
u_n	velocity on boundary surface in direction of unit surface normal vector	μ
V	geometric cell volume (i.e. $\Delta x \cdot \Delta y \cdot \Delta z$ or $\Delta r \cdot r \cdot \Delta \theta \cdot \Delta z$)	π
V_f	fluid volume in cell (i.e. $\gamma_v \cdot V$)	ρ
v	velocity component in y (or θ) direction (i.e. u_y or u_θ)	σ_n
w	velocity component in z direction (i.e. u_z)	τ_n
X	radian value indicating pump operating state in $\tilde{Q}$ - $\tilde{N}$ plane	ϕ
x	i-direction spatial coordinate in Cartesian system	Ω_n
x_M	spatial coordinate in direction M, where M is x or r, y or θ , or z (general)	ω_n
y	j-direction spatial coordinate in Cartesian system	
z	k-direction spatial coordinate in Cartesian and cylindrical systems	<u>Subs</u>
γ_M	directional surface porosity in direction M, where M is x or r, y or θ , or z	b
γ_v	volume porosity	cool
ΔP	pressure difference between two spatial locations	f
$\Delta r, \Delta \theta$	mesh spacing in r and θ directions, respectively, for cylindrical system	gap
Δt	time step size	i
$\Delta x, \Delta y$	mesh spacing in x and y directions, respectively, for Cartesian system	in
Δz	mesh spacing in z direction for Cartesian and cylindrical systems	j
δ_ϕ	residual of conservation equation for quantity ϕ	k
ε	turbulent kinetic energy dissipation rate	ℓ
ε_L	fraction of black-body radiation which is emitted by radiation surface number L ("emissivity" in Sec. 5.4)	
ε_n	constants in convergence criteria (Sec. 12.3 and 12.4)	m
θ	j-direction spatial coordinate in cylindrical system ("angle")	p

thermal conductivity

viscosity

ratio of circumference to diameter for a circle (i.e. 3.14159...)

density

constants in turbulence model, where n is h , k , and ε

pump torques, where n is f (friction), h (hydraulic), and m (motor)

unknown variable of solution (i.e. h , k , P , u , v , w , or ε) in general conservation equation

relaxation parameters (Sec. 12.4)

relaxation parameters (Sec. 12.4)

sg:

boundary

coolant (omitted unless essential)

fluid (omitted unless essential)

gap between adjacent material regions of a thermal structure element

cell location index in x (or r) spatial direction

inlet

cell location index in y (or θ) spatial direction

cell location index in z spatial direction

laminar component (used with viscosity (μ), thermal conductivity (λ), or friction factor (f)); also, fluid cell location relative to a base cell, where 0 is base cell, 1 is cell on $-i$ side, 2 is cell on $+i$ side, 3 is cell on $-j$ side, 4 is cell on $+j$ side, 5 is cell on $-k$ side, and 6 is cell on $+k$ side (used in general form of the conservation equations)

fluid cell number

pump

R	reference
r	radiation surface
ref	reference
s	solid or sink
ss	steady state
t	turbulent component (used with viscosity (μ), thermal conductivity (λ), or friction factor (f))
w	wall material
0	initial or reference

Superscripts:

i	inner iteration number
n	time step number
o	outer iteration number
ϕ	conservation equation to which coefficient belongs (general)

Miscellaneous:

$ \alpha $	absolute value of α
$\bar{\alpha}$	"bar" denotes an averaging process is used to define α
$\tilde{\alpha}$	"tilde" denotes α is a dimensionless quantity (e.g. $\alpha/\alpha_{\text{ref}}$)
$\vec{\alpha}$	"arrow" denotes α is a vector quantity
$d\alpha/d\beta$	derivative of α with respect to β
$\partial\alpha/\partial\beta$	partial derivative of α with respect to β
$\sum_{n=1}^N$	summation over n from 1 to N

>| upwind value, which is either ϕ_j or ϕ_i depending on whether the multiplying factor is positive or negative, respectively

An index of the FORTRAN variable names (e.g. ISTATE, IMAX, JMAX, KMAX) used in the various NAMELISTs in the card-image input file is provided as Appendix B of the "COMMIX-1AR/P Input Guide", which is included as Appendix A of this report.

ABSTRACT

The COMMIX-1AR/P computer program is designed for analyzing the steady-state and transient aspects of single-phase fluid flow and heat transfer in three spatial dimensions. This version is an extension of the modeling in COMMIX-1A to include multiple fluids in physically separate regions of the computational domain, modeling descriptions for pumps, radiation heat transfer between surfaces of the solids which are embedded in or surround the fluid, a $k-\epsilon$ model for fluid turbulence, and improved numerical techniques. The porous-medium formulation in COMMIX allows the program to be applied to a wide range of problems involving both simple and complex geometrical arrangements. The input preparation and execution procedures are presented for the COMMIX-1AR/P program and several postprocessor programs which produce graphical displays of the calculated results.

EXECUTIVE SUMMARY

The COMMIX (COMponent MIXing) computer programs are designed for analyzing steady-state and transient aspects of fluid flow and heat transfer in three spatial dimensions. Program versions in this series having a "1" in the suffix are for single-phase fluids having limited compressibility.

The COMMIX-1AR/P version is an extension of the COMMIX-1A version to include:

- allowance for multiple fluids in physically separate regions,
- radiation heat transfer between surfaces of solids embedded in or surrounding the fluid,

- a k- ϵ model for fluid turbulence, and
- improved numerical methods.

Numerous second-order changes have been implemented to increase the user's flexibility in specifying boundary conditions and material properties and in utilizing a wide range of modern mainframe computers.

The three-dimensional modeling in COMMIX is achieved using a porous-medium formulation. This has been implemented using the concepts of volume porosity (the fraction of a control volume occupied by fluid), directional surface porosity (the fraction of a control volume surface through which fluid may flow), distributed resistances to fluid flow, and distributed heat sources. Either Cartesian or cylindrical coordinates may be used. The fluid conservation equations for mass, momentum, and energy and the transport equations for the turbulence parameters are put into finite-difference form by averaging over the local control volumes in the system. This process allows simulations of simple problem geometries (e.g. straight pipe) and complex problem geometries (e.g. fuel bundle, heat exchanger, reactor vessel). Although primarily developed for nuclear reactor applications, the models in COMMIX are sufficiently general to be used for analyzing other engineering systems.

The material properties for the fluids and embedded solids, heat transfer coefficients, and friction factors are assumed to have various basic forms to which the user supplies coefficients. In some cases, the user may supply the coding to override these basic forms. Detailed fluid property packages are included for liquid sodium, liquid water (both normal and heavy), and helium gas, as well as a library of friction factors for specific applications.

The discretized conservation equations are solved as an initial-value problem in time and a boundary-value problem in space. The user is provided with a broad range of boundary condition choices, allowing the simulation of the usual types of boundary conditions encountered in real systems. The boundary conditions and embedded heat sources may be functions of time. The formulation of the conservation equations is implicit in time.

A previous report presented the modeling features (covering the basic equations and underlying assumptions) and the numerical methods (covering spatial differencing, implicitness of temporal differencing, and iterative solution methods). Given this background, the present report concentrates on explaining how to use the computer program, covering input preparation and program execution. Additional reports describe the internals of the computer program and provide assessments of the modeling.

1. INTRODUCTION

The COMMIX-1AR/P computer program is intended for steady-state and transient analysis of fluid flow and heat transfer in three dimensions. Although primarily developed for nuclear reactor applications, the COMMIX-1AR/P computer program can be used to analyze the fluid behavior in other engineering systems.

A previous report (Ref. 1) described the features of COMMIX-1AR/P, including the specific equations being solved in each area of the modeling. The user of the program is assumed to have read this prior report and have it available as a reference when using the program.

The present report focusses on how to use the code, describing the input which must be specified by the user for each aspect of the modeling performed by the program. Section 2 provides a general description of the input data formats used by the program. The remaining sections in the main part of this report discuss the input required by model category. All cases require input associated with geometry (as described in Sec. 3), boundary and initial conditions (Sec. 4), and various auxiliary input (Sec. 12). Depending on the problem being modeled, the user may need to specify input for thermal structures (Sec. 5), force structures (Sec. 6), pumps (Sec. 7), fluid turbulence (Sec. 8), fluid properties, structure properties, and heat transfer coefficients (Sec. 9), hexagonal fuel assemblies (Sec. 10), and mass rebalancing (Sec. 11).

As discussed in Sec. 2, each model type usually requires input in more than one block of the input file. As an aid to input preparation, the main part of Appendix A provides a block-by-block description of the input file. Included in this description is the default value for each input variable. This appendix also contains a listing of all error messages produced by the program and the actions required by the user to eliminate the error. This appendix is distributed in machine-readable form with the program, to facilitate its dissemination to multiple users at a site. The telephone numbers and mailing addresses of the authors are provided at the beginning of Appendix A for users having questions or comments about the use of the program.

The other appendices are as follows. Appendix B provides the equation-of-state description for the four sets of detailed fluid properties which are supplied with the program. Appendix C describes how to install and execute the COMMIX-1AR/P program on your system, providing specific sample installation and execution jobstreams for both IBM and Cray mainframe environments. The various graphics postprocessor programs distributed with COMMIX-1AR/P are described in Appendix D. A sample case, with input and output, is provided in Appendix E.

Those who have used COMMIX-1 (Ref. 2), COMMIX-1A (Refs. 3 and 4), or COMMIX-1B (Refs. 5 and 6) should have little difficulty converting their input to run on COMMIX-1AR/P or developing input for new cases -- much of the input is essentially unchanged from these prior program versions. The user is, however, cautioned that the input for COMMIX-1AR/P is not an upward-compatible transition from any one of these prior COMMIX versions; some old input variables have been deleted, some old input variables have new meanings, and some new input variables have been added; these changes were necessitated by the differences in model features between the various versions.

2. INPUT FILES AND DATA FORMATS

There are two operational modes for COMMIX: steady state and transient. The program is solving the same set of equations in each mode, but there are differences in input. Each mode of operation may take more than one computer run to complete the calculation, i.e., to converge to steady-state conditions or to complete the desired transient history. The input for the first run in either a steady-state or a transient calculation differs from the input required in successor runs of the same problem. A transient calculation should normally be started using initial conditions calculated by running COMMIX in the steady-state mode.

There are two types of user-supplied input to COMMIX: card image on FORTRAN unit number 5 and binary on unit number 9. The contents of the card-image file and whether the binary file is used are functions of the calculational mode status. This is indicated by the user-input variable ISTATE. The four values of ISTATE and their meanings are shown in Table 2.1. The operational philosophy in COMMIX is such that all data are input by the user at ISTATE=0, except for the transient functions describing boundary conditions which are input at ISTATE=2. The user is generally restricted to only inputting control information (time step size, maximum number of iterations, printing specifications, etc.) at ISTATE=1 and ISTATE=3. The majority of the input values can not be changed at intermediate points in the calculation (i.e. ISTATE>0); instead, data are passed from one computer run to its successor run via the binary file, which is denoted the "restart" file.

Table 2.1 Definition of ISTATE¹

ISTATE	Meaning
0	First computer run in a steady-state condition
1	Continuation of a steady-state calculation
2	First computer run in a transient calculation
3	Continuation of a transient calculation

Notes:

1. The value of ISTATE is input in NAMELIST DATA.

The card-image input to COMMIX consists of a number of different blocks. The ordering of these blocks is shown in Table 2.2 along with the restrictions on whether the block is required (denoted by R), optional (denoted by O), or disallowed (denoted by N) for the four values of ISTATE. The first block in the card-image input file is a set of problem description records; the user may include any number of 80-character records which describe the case; these records will be merely copied to the print output file (along with the rest of the card-image input file). The rest of the card-image input file consists of NAMELIST and fixed-format blocks. The various models in COMMIX generally require input values in NAMELIST GEOM and/or DATA plus one or two other blocks. The specific variables in each input block are described in Appendix A.

Table 2.2 Ordering of Blocks in the Card-Image Input File

Block Type	ISTATE ¹				Notes ¹
	0	1	2	3	
Problem Description	O	O	O	O	
NAMELIST GEOM	R	RL	RL	RL	
Boundary Surface Identification	R ²	N	N	N	E
NAMELIST DATA	R	RL	RL	RL	
NAMELIST INPUTQ	O	N	*L	N	
Rebalancing Regions	O	N	O	N	E
Fluid Cell Locations	O	N	N	N	E
NAMELIST PUMP	O	N	*L	N	
Pump Locations	O	N	*L	N	E
Force Structure Locations	O	N	O	N	E
NAMELIST STRUCT	O	N	O	N	
Thermal Structure Prototypes	O	N	O	N	E
Thermal Structure Locations	O	N	O	N	E
Boundary Value Initializations	R	RL	RL	RL	E
Internal Cell Initializations	R	RL	RL	RL	E

Notes:

- Key to codes used under ISTATE and Notes in above table:
E: An 'END' record is part of and terminates this block.
L: Limited input allowed.
N: Not allowed.
O: Optional (depending on other flags in input stream).
R: Required.
*: If block was present at ISTATE=0, it must be present at ISTATE=2 (even if there are no changes).
- Required for IGEOM ≤ 0; not allowed for IGEOM > 0.

The format of a NAMELIST block must follow certain rules that are somewhat machine dependent, since the NAMELIST construct is not part of the FORTRAN-77 ANSI Standard (Ref. 7). The following example illustrates the basic format structure of a NAMELIST block:

```
&DATA  ISTATE=2,  DT=2*.01,.0001,100.,
      KPRES(10)=101, KFLOW(9)=102,
      TVAL=0.,1.,10.,100.,  0.,1.,10.,100.,
      FVAL=1.,.5,.2,.1,      1.,.9,.1,.01,
      NEND=4,                  4,          &END
```

A NAMELIST block consists of 80-character records. Position 1 of every record in the NAMELIST block must be blank. The block begins with an ampersand (&) followed by the block name (which is "DATA" in this example) followed by a blank and is terminated by "&END". Between these beginning and terminating block delimiters, the entries have the form "VARIABLE_NAME=value_list,". The VARIABLE_NAME is either (1) a simple variable name (e.g. ISTATE), (2) an array name (e.g. DT, TVAL, FVAL, NEND), or (3) a specific array element (e.g. KPRES(10), KFLOW(9)). The format of the value_list depends on the VARIABLE_NAME and is either (1) a single constant for a simple variable name (e.g. ISTATE=2,) or a specific array element (e.g. KPRES(10)=101,) or (2) a list of constants separated by commas (,) for an array name (e.g. NEND=4,4,); in this latter case, the values specified are used to fill the array starting with element 1, and the value_list may utilize the format "n*value,", where n is an INTEGER constant, to indicate repetition of the value n times (e.g. DT=2*.01,.0001,100.,). The value_list is separated from the VARIABLE_NAME by an equals sign (=) and is followed by a comma (,). The constants in the value_list must agree in type with the VARIABLE_NAME; in this regard, most input variables in COMMIX follow the implicit typing in FORTRAN that variable names beginning with the letters I, J, K, L, M, or N are INTEGER and all others are floating point (i.e. REAL on long-word machines and DOUBLE PRECISION on short-word machines); the few exceptions to this typing rule will be specifically noted in the input instructions as they occur. Floating-point constants may be specified either with or without exponents (i.e. 1.2345E+2 and 123.45 are equally acceptable). The VARIABLE_NAMES and the constants in the value_list must not contain embedded blanks; one or more blanks may optionally precede or follow the commas used as separators. This pattern of "VARIABLE_NAME=value_list," is repeated for all variables which need to be specified. Within a NAMELIST block, the variable names may appear in any order. The user

should consult the FORTRAN Reference Manual for the machine on which COMMIX is being executed for any system-imposed restrictions on NAMELIST input formats; for example, (1) some systems allow the final comma following the value_list to be omitted and (2) the IBM and Cray allow the block name (except as noted in the next sentence) and VARIABLE_NAME fields in a NAMELIST block to be specified using either upper- or lower-case letters; the examples in this report show the final comma and use upper-case letters. In addition to the system-imposed restrictions, COMMIX imposes additional restrictions on NAMELISTs T, F, and M in the Thermal Structure Prototype block: the ampersand before the block name must be in column 2 and the block name (in column 3) must be an upper-case letter.

The primary advantage of using the NAMELIST construct is that the user need only specify values for variables which are to be changed from their "current values". The "current value" of an input variable depends on the value of ISTATE: for ISTATE=0, the "current value" is the program-defined default value, which is shown in Appendix A for each input variable; for ISTATE>0, the "current value" is the value read from the restart file. When preparing input for ISTATE>0, the user should generally only include specifications for variables which are to be assigned values which differ from their current values. If a NAMELIST block is required to be present in the card-image input file but the user does not wish to specify any variables as having values different from their current values, an empty NAMELIST of the form

```
&name  &END
```

must be entered (where "name" is to be replaced with an actual NAMELIST block name) rather than omitting the NAMELIST block entirely.

The other type of card-image input block is fixed format. The records in each block consist of a 4-character keyword, followed by a 4-digit integer value or a 10-digit floating-point value, followed by six or seven 4-digit integer indices denoting a spatial location. The specific formats, keywords, values, and index meanings are described for each block in Appendix A. All letters in the keyword field must be typed as upper case. These blocks are terminated by a record

containing 'END ' in the first four positions of a record; this terminator record is considered part of the block and must be present if the block is required even if no other records are present in the block.

The user may include comments in several places in the card-image input file. All records in the Problem Description block (i.e. everything up to NAMELIST GEOM) are treated as comments. Any information in a record following the "&END" terminator of a NAMELIST block is treated as a comment. The last portion of each record in a fixed-format block is a comment field. Comment records may be included between the end of a NAMELIST or fixed-format block and the beginning of the next NAMELIST block as long as (1) it does not start with _&GEOM_, _&DATA_, _&INPUTQ_, _&PUMP_, or _&STRUCT_ and (2) character positions 1 through 4 do not contain one of the following strings:

&F	_&M_	_&T_	AL__	ALX_	ALY_	ALZ_
AREA	END_	HL__	HLB_	HTEX	IN__	IREG
OUT_	P__	PB__	PUMX	PUMY	PUMZ	QBN_
QSOU	REBM	REBX	REBY	REBZ	REG_	RLB_
TL__	TLB_	TURC	TURV	UL__	VELB	VL__
WL__	XFOR	YFOR	ZFOR			

where the underscore () indicates a blank. On the other hand, comments may not be embedded within a NAMELIST block (except on the Cray, where the part of a record following a colon (:) or semicolon (;) is taken as a comment) or within a fixed-format block (except within the comment portion of a record). The user will normally want to use these comment opportunities to annotate the input deck.

All user input is provided by the card-image file at ISTATE=0. When continuing the same calculation at ISTATE>0, minimal user input is provided by the card-image file; instead, the bulk of the data is provided by the binary-format restart file. This file is written at the end of a COMMIX computer run (cf. Sec. 12.1) and contains sufficient data for the calculation to be continued in a later computer run.

The specific contents of the card-image file will be discussed by model type in the various sections of this report; a complete description of this file is given in Appendix A. The way in which the card-image and restart files are passed to COMMIX is system dependent; several examples are given in Appendix C.

During execution, COMMIX produces a printable output file on FORTRAN unit number 6, a binary-format restart file on unit number 10, and a binary-format plot file on unit number 76. The user input which controls the information on these files is discussed in Sec. 12.6 and examples of the usage of these files are discussed in Appendix C.

The units used for floating-point quantities are shown in square brackets throughout this report. These units generally conform to the International System of Units (SI), having the base units of meter [m] for length, kilogram [kg] for mass, and second [s] for time. There are two exceptions to this standard: (1) all temperatures are in degrees Celsius [$^{\circ}\text{C}$] rather than kelvins [K] and (2) pump speed is in revolutions per minute [rev/min] rather than radians per second [s^{-1}]. The notation [-] denotes that a floating-point quantity is dimensionless. All fixed-point quantities are dimensionless, and the [-] notation is omitted.

3. GEOMETRY

The first step in modeling a physical system using COMMIX is definition of the geometry. This section describes the elements involved in defining the geometry and how the user passes this geometry definition to COMMIX. The fluid cell geometry described in this section is input only at the beginning of a steady-state calculation (i.e. ISTATE=0).

3.1 Grid Definition

There are three choices for the overall system geometry: Cartesian, cylindrical, and hexagonal fuel bundle. The geometry selection is indicated by user-input variable IGEOM in NAMELIST GEOM. As shown in Table 3.1, a Cartesian coordinate system with spatial coordinates (x,y,z) is selected by IGEOM=0 and a cylindrical coordinate system with spatial coordinates (r,θ,z) is selected by IGEOM=-1. The hexagonal fuel pin bundle geometry is selected by IGEOM > 0 and having a value equal to the number of fuel pins in the bundle (where allowed values are 7, 19, 37, 61, 91, 127, 169, 217, and 271); specifying IGEOM > 0 results in selection of a Cartesian coordinate system and automates many aspects of the geometry specification process. The majority of this section will concentrate on the basic Cartesian and cylindrical coordinate system geometries. The specific details of the hexagonal fuel assembly geometry will be covered in Sec. 10.

The system volume must be divided into "computational cells" by the user. The computational cells are the "main control volumes" defined in the Equations and Numerics report (Ref. 1). This process is illustrated in Fig. 3.1. The location of the faces of computational cells are determined by user-specified spacings along each of the three coordinate directions. The specific variables to be input by the user are all in NAMELIST GEOM and are shown in Table 3.1. The number of cell spacings in the x or r direction is input using variable IMAX, and the cell spacings are input using array DX(I), for $I=1,2,\dots,IMAX$. The number of cell spacings in the y or θ direction is input using variable JMAX, and the cell spacings are input using array DY(J), for $J=1,2,\dots,JMAX$; the units for the cell spacing in array DY are meters for Cartesian coordinates (IGEOM ≥ 0) and radians for cylindrical coordinates (IGEOM = -1).

Table 3.1. Grid Definition Variables in NAMELIST GEOM

Variable	Meaning/Value	Restrictions
IGEOM	select coordinate system geometry; may be: -1: cylindrical (r,θ,z); 0: Cartesian (x,y,z); >0: hexagonal fuel-pin bundle in Cartesian coordinates.	IGEOM=7,19,37,61,91,127,169,217,271
IMAX	maximum number of cells in the x- or r-direction.	$1 \leq \text{IMAX} \leq 99$
JMAX	maximum number of cell in the y- or θ-direction.	$1 \leq \text{JMAX} \leq 99$
KMAX	maximum number of cells in the z-direction.	$1 \leq \text{KMAX} \leq 99$
NM1	total number of cells.	$\text{NM1} \leq \text{IMAX} * \text{JMAX} * \text{KMAX}$
DX(I)	Δx or Δr : cell spacing [m] in the x- or r-direction.	$\text{DX(I)} > 0$ $1 \leq I \leq \text{IMAX}$
DY(J)	Δy or $\Delta \theta$: cell spacing [m or radian] in the y- or θ-direction.	$\text{DY(J)} > 0$. $1 \leq J \leq \text{JMAX}$ If IGEOM=-1: $\sum_{J=1}^{\text{JMAX}} \text{DY(J)} \leq 2\pi$
DZ(K)	Δz : cell spacing [m] in the z-direction.	$\text{DZ(K)} > 0$. $1 \leq K \leq \text{KMAX}$

The number of cell spacings in the z direction is input using variable KMAX, and the cell spacings are input using array DZ(K), for K=1,2,...,KMAX. The total number of fluid cells must be input using variable NM1; the value input may be an estimate which is larger than the actual number of fluid cells; the actual number of fluid cells will be printed by the code during input processing.

A fully three-dimensional geometry has all three of IMAX, JMAX, and KMAX greater than 1; a two-dimensional geometry has one of these three equal to 1; a one-dimensional geometry has two of these three equal to 1. There is a tradeoff involved in selecting the number of cell spacings: there must be sufficient cells to resolve the flow field in the regions of interest but the

(a)

(b)

Fig. 3.1. Partitioning of Flow Domain into Computational Cells for Two Coordinate Systems: (a) Cartesian and (b) Cylindrical

computer memory and CPU time required go up as more cells are used. The cell spacings are normally chosen so that cell face locations correspond to significant features within the system, e.g. the location of an inlet pipe or a flow baffle plate. The cell spacings may be nonuniform, both within a given direction and from one direction to another; however, the approximations inherent in the finite-difference approach used in COMMIX become less accurate as the cell spacing becomes less uniform. The user should also take advantage of existing physical symmetry conditions, which may allow the COMMIX geometry to only model a half, a third, a quarter, etc. of the system. In this regard, there is no requirement for modeling the full 2π geometry when using cylindrical coordinates.

3.2 Grid Orientation

The absolute orientation of the grid system is established by the gravitational acceleration vector ($\vec{g}$), which is defined by

$$\vec{g} = g_x \vec{i}_x + g_y \vec{i}_y + g_z \vec{i}_z \quad (3.1a)$$

in Cartesian coordinates and by

$$\vec{g} = g_r \vec{i}_r + g_\theta \vec{i}_\theta + g_z \vec{i}_z \quad (3.1b)$$

or

$$\vec{g} = [\cos(\theta)g_x + \sin(\theta)g_y] \vec{i}_r + [-\sin(\theta)g_x + \cos(\theta)g_y] \vec{i}_\theta + g_z \vec{i}_z \quad (3.1c)$$

in cylindrical coordinates, where $\vec{i}$ is the unit vector in the spatial direction indicated by its subscript. The user defines the gravitational acceleration vector by inputting g_x , g_y , and g_z using the input variables GRAVX, GRAVY, and GRAVZ, respectively, in NAMELIST DATA as shown in Table 3.2. These components are entered in Cartesian coordinate form even when a cylindrical coordinate system is being used; the code internally computes g_r and g_θ , which are a function of the angular position θ , from the input values of g_x and g_y ; in this regard, the x axis in Cartesian coordinates corresponds to the r axis at $\theta=0$ in cylindrical coordinates, the y axis

Table 3.2. Grid Orientation Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
GRAVX	g_x : x-component of gravitational acceleration vector [m/s ²].	
GRAVY	g_y : y-component of gravitational acceleration vector [m/s ²].	
GRAVZ	g_z : z-component of gravitational acceleration vector [m/s ²].	

in Cartesian coordinates corresponds to the r axis at $\theta = \pi/2$ in cylindrical coordinates, and the z axis is the same in both coordinate systems. The magnitude of the gravitational acceleration vector (i.e. $(g_x^2 + g_y^2 + g_z^2)^{0.5}$) should correspond to the actual gravitational acceleration (nominally 9.8 m/s²). The sign is significant when defining the components of the gravitational acceleration vector. By way of example, the user would input GRAVX=0.0, GRAVY=0.0, and GRAVZ=-9.8 to have the positive z axis of the coordinate grid pointing directly upward.

3.3 Fluid Assignment

Each computational cell contains one single-phase fluid. The number of distinct fluids used in a calculation is input using variable IFHTX in NAMELIST GEOM as shown in Table 3.3.

Table 3.3. Fluid Use Variable in NAMELIST GEOM

Variable	Meaning/Value	Restrictions
IFHTX	number of fluids being used.	$1 \leq \text{IFHTX} \leq \text{PIFHTX}$

The default value for IFHTX is 1; the value input for IFHTX must not exceed the value of code-internal variable PIFHTX; typically, PIFHTX=5 but this may have been altered by the person who installed COMMIX at your site (cf. Appendix C.1.1). As a default, all computational cells are assumed to contain fluid number 1. In a multifluid calculation, the user must specify which computational cells contain fluid numbers 2 through IFHTX in order to override the default fluid number specification. This is accomplished using the Fluid Cell Location input block. The

Fluid Cell Location block consists of a set of fixed format records having the record structure shown in Table 3.4. As these records are processed, the value of NUM is stored in array INDHTX to override the default fluid number of 1 for the indicated cells in the region of the system volume delimited by indices IB, IE, JB, JE, KB, and KE. The user must place a separation barrier between computational cells which contain different fluid numbers; this is accomplished using either a pair of boundary surfaces or by a zero-porosity cell face; both of these concepts are discussed later in this section.

Table 3.4. Fluid Cell Location Input Block Record Structure

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier; may be 'HTEX': "NUM" is the fluid number; 'END ': terminates this input block; record positions 5-32 should be blank.	(see below)
5-8	I4	NUM	see NAME.	$2 \leq \text{NUM} \leq \text{IFHTX}$
9-12	I4	IB	beginning I-range index.	$1 \leq \text{IB} \leq \text{IE}$
13-16	I4	IE	ending I-range index.	$\text{IB} \leq \text{IE} \leq \text{IMAX}$
17-20	I4	JB	beginning J-range index.	$1 \leq \text{JB} \leq \text{JE}$
21-24	I4	JE	ending J-range index.	$\text{JB} \leq \text{JE} \leq \text{JMAX}$
25-28	I4	KB	beginning K-range index.	$1 \leq \text{KB} \leq \text{KE}$
29-32	I4	KE	ending K-range index.	$\text{KB} \leq \text{KE} \leq \text{KMAX}$
33-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. These records result in assigning the value "NUM" to the variable named "INDHTX" for the spatial positions (I,J,K) in the three-dimensional region delimited by $\text{IB} \leq \text{I} \leq \text{IE}$, $\text{JB} \leq \text{J} \leq \text{JE}$, and $\text{KB} \leq \text{K} \leq \text{KE}$ within the system volume.
2. Use as many 'HTEX' records as needed to assign nondefault fluid number; the order is irrelevant except (a) if a cell is assigned a value in more than one record, the final such specification will be used and (b) the 'END ' record only occurs once and is last.
3. The indices (I,J,K) refer to cell-center grid points; the cells must actually exist (i.e. contain fluid).

3.4 Boundary Surface Definition

A grid point is located at the geometric center of each computational cell. Most fluid quantities (i.e density, pressure, enthalpy, temperature, turbulent kinetic energy, and turbulent kinetic energy dissipation) are defined at the grid points; however, the three velocity components are defined at the centers of the three cell faces. A computational cell which is surrounded only by other computational cells is called an "internal cell". If one or more of the cell faces is on a boundary surface (see below), then the cell is called a "boundary cell".

The set of computational cells forming the system volume must be completely enclosed by a set of "boundary surfaces". These boundary surfaces are nonmoving but may be open or closed to the flow of mass and energy; they may represent true walls or planes of symmetry. A boundary surface is defined by the orientation of the unit surface-normal vector, $\vec{n}$, which points from the surface toward the fluid:

$$\vec{n} = n_x \vec{i}_x + n_y \vec{i}_y + n_z \vec{i}_z \quad (3.2a)$$

in Cartesian coordinates or

$$\vec{n} = n_r \vec{i}_r + n_\theta \vec{i}_\theta + n_z \vec{i}_z \quad (3.2b)$$

in cylindrical coordinates, where $\vec{i}$ is the unit vector in the coordinate direction indicated by the subscript. There are two types of boundary surfaces: regular and irregular. A regular boundary surface has its unit surface-normal vector aligned with one of the three coordinate directions; the system volume shown in Fig. 3.2 is completely enclosed using regular boundary surfaces; the components of the unit surface-normal vectors are shown in Table 3.5 for each boundary surface. An irregular boundary surface has no restrictions on the alignment of the unit surface-normal vector; surface number 1 of the system volume shown in Fig 3.3 is an irregular surface; the components of the unit surface-normal vectors are shown in Table 3.6 for each boundary surface. (The geometries depicted in Fig. 3.2 and 3.3 also show that the total number of fluid cells, NM1, may be less than the product of the maximum number of cells in the three spatial directions, IMAX*JMAX*KMAX.)

Table 3.5. Unit Surface-Normal Vectors for Figure 3.2

Surface Number	n_x	n_y	n_z
1	0.	1.	0.
2	-1.	0.	0.
3	0.	0.	1.
4	1.	0.	0.
5	0.	0.	-1.
6	0.	0.	-1.
7	0.	-1.	0.
8	-1.	0.	0.

Table 3.6. Unit Surface-Normal Vectors for Figure 3.3

Surface Number	n_x	n_y	n_z
1	-0.707	0.	-0.707
2	1.	0.	0.
3	0.	1.	0.
4	0.	-1.	0.
5	0.	0.	1.
6	-1.	0.	0.
7	0.	0.	-1.

A section of a boundary surface between two pairs of consecutive grid lines is defined as a boundary surface element. A boundary surface may have more than one boundary surface element. For the example shown in Fig. 3.2b, boundary surface number 6 has nine boundary surface elements. A boundary surface element is thus the face of a computational cell which is coincident with the boundary surface. Boundary surfaces are, in fact, located by identifying all such adjacent computational cells toward which the unit surface-normal vector points. A computational cell may have several of its faces which are boundary surface elements; however,

Fig. 3.2. Example Geometry Showing (a) Boundary Surface Numbering and (b) Partitioning into Boundary Surface Elements

(a)

(b)

Fig. 3.3. Example Geometry Having an Irregular Boundary Surface: (a) Surface Numbering and (b) Partitioning into Surface Elements

only one such face of a cell may be a boundary surface element on any irregular boundary surface.

Each boundary surface must be assigned a boundary surface number by the user. The same boundary surface number may be assigned to all boundary surfaces which have (1) the same unit surface-normal vector, (2) the same velocity, energy, and pressure boundary conditions (cf. Sec. 4.1), and (3) adjacent cells containing the same fluid number. The example in Fig. 3.2 has been shown with 8 boundary surfaces; in this case, boundary surfaces 2 and 8 have the same unit surface-normal vector but are in different planes. If the boundary conditions and adjacent fluid number were also the same, then these two boundary surfaces could be combined into one boundary surface as shown in Fig. 3.4. Conversely, sometimes a given physical surface must be divided into two different boundary surfaces; for the example in Fig. 3.5, the top surface has been divided into boundary surface 5, which is closed to flow, and boundary surface 6, which is open to flow. There are three further restrictions on boundary surface numbering: (1) the numbering must be contiguous (i.e. $N=1,2,3,\dots,NSURF$), (2) all irregular boundary surfaces must have boundary surface numbers which are less than the boundary surface numbers assigned to regular surfaces, and (3) there is a limit of 99 boundary surfaces.

A Cartesian coordinate geometry needs a minimum of six boundary surfaces: one having its unit surface-normal vector in the plus direction and one having its unit surface-normal vector in the minus direction for each of the three coordinate directions. The same is true in cylindrical coordinates when something less than the full 2π geometry is modeled. If the point $r=0$ is to be included in a cylindrical geometry problem, then a boundary surface must be defined at this position; it will have zero surface area and should be assigned symmetry-type boundary conditions. When the full 2π geometry is modeled in cylindrical coordinates, a minimum of four boundary surfaces is needed: one having its unit surface-normal vector in the plus direction and one having its unit surface-normal vector in the minus direction for the r and z directions; in the theta direction, cells at $J=JMAX$ are automatically linked to cells at $J=1$ rather than the fluid being contained by boundary surfaces. System volumes typically need more than these minimum number of boundary surfaces in order implement the desired boundary conditions. If the system control volume contains multiple fluids, then a pair of boundary surfaces should

Fig. 3.4 Example of Noncoincident Surfaces Having the Same Boundary Surface Number (2)

Fig. 3.5. Examples of Coincident Surfaces Having Different Boundary Surface Numbers (6 vs. 5 and 8 vs. 3)

be used to separate the two fluids regions from each other; the two boundary surfaces are at the same physical location but have unit surface-normal vectors pointing in opposite directions (i.e. one pointing toward each fluid). Boundary surfaces may also be used to delineate regions inside of the bulk system volume in which there is no fluid; an example of this type of geometry is shown in Fig. 3.6.

The geometrical description of boundary surfaces is in three blocks of the card-image file: NAMELIST GEOM, NAMELIST DATA, and Boundary Surface Identification. These input variables are discussed below. The input variables associated with boundary conditions will be covered in Sec. 4.1.

The boundary surface input variables in NAMELIST GEOM are shown in Table 3.7. The total number of surfaces is input using variable NSURF. For each boundary surface, the components of the unit surface-normal vector, as used in Eq. 3.2, are input using arrays XNORML(N), YNORML(N), and ZNORML(N), for $N=1,2,\dots,NSURF$; the signs of these components are important: the unit vectors point from the boundary surface toward the fluid; the user must input properly normalized components as noted in the table; for a regular (as opposed to irregular) boundary surface, the entry in one of these arrays is ± 1.0 and the entries in the other two arrays are 0.0. The total number of boundary surface elements, summed over all boundary surfaces, is input using variable NL1; the value input may be an estimate which is larger than the actual number of boundary surface elements; the actual number of boundary surface elements will be printed by the code during input processing. Nonzero values of the other two input variables shown in Table 3.7 produce various forms of debug output during the input processing stage. The boundary surface summary printed for $IBSBUG > 0$ can be useful during the early stages of checking the boundary surface input. The additional printing of cell and surface element number arrays by $LMPRNT > 0$ is not generally useful for the normal user.

The one boundary surface input variable in NAMELIST DATA is shown in Table 3.8. The entries in array KPROP(N), for $N=1,2,\dots,NSURF$, indicate which fluid, by index number, is in the cells immediately adjacent to each boundary surface and pointed to by the unit surface-normal vector.

(a)

(b)

Fig. 3.6. Examples Showing Partitioning of Complex Geometries

Table 3.7. Boundary Surface Definition Variables in NAMELIST GEOM

Variable	Meaning/Value	Restrictions
NSURF	number of unique surfaces.	$4 \leq \text{NSURF} \leq 99$
NLI	total number of surface elements.	
XNORML(N)	n_x or n_r : x- or r-component of unit surface-normal vector for surface N.	$1 \leq N \leq \text{NSURF}$ (see below)
YNORML(N)	n_y or n_θ : y- or θ -component of unit surface-normal vector for surface N.	$1 \leq N \leq \text{NSURF}$ (see below)
ZNORML(N)	n_z : z-component of unit surface-normal vector for surface N.	$1 \leq N \leq \text{NSURF}$ (see below)
IBSBUG	select printing of Boundary Surface Summary, which is a compact "graphical" rendering on printer of all boundary surface locations; may be: 0: summary is not printed; 1: summary is printed and execution continues; 2: summary is printed and execution terminates.	
LMPRNT	select printing of cell and surface number arrays; these arrays are of little use to the normal user; may be: 0: neither array is printed; 1: cell number array is printed and execution terminates; 2: cell and surface number arrays are printed and execution terminates.	

Notes:

1. Surface-normal vectors point from surface toward fluid.
2. $\text{XNORML}(N)^2 + \text{YNORML}(N)^2 + \text{ZNORML}(N)^2 = 1$.

Table 3.8. Boundary Surface Definition Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
KPROP(N)	fluid number in cells adjacent to surface N.	$1 \leq \text{KPROP}(N) \leq \text{IFHTX}$ $1 \leq N \leq \text{NSURF}$

The specific locations of the boundary surfaces are identified using a set of fixed format records in the Boundary Surface Identification input block. The record structure of this input block is shown in Table 3.9. The first field in the record indicates the type of boundary surface: 'IREG' indicates an irregular boundary surface and 'REG' indicates a regular boundary surface. The second field in the record is the area of each boundary surface element being defined using this record; this must be explicitly specified for irregular surfaces; the user may accept an internally generated geometric value by entering a negative value for regular boundary surfaces. The third through eighth fields in the record specify a range of one or more cell locations which are immediately adjacent to the boundary surface and toward which the unit surface-normal vector points. The ninth field in the record specifies the boundary surface index number. All records associated with boundary surface N must be grouped together and immediately follow those associated with boundary surface N-1. Irregular boundary surfaces must have index numbers which are lower than the index numbers used for regular boundary surfaces. An 'END' record must be used to terminate this input block.

3.5 Volume and Surface Porosities

The fluid conservation equations in COMMIX are all formulated on a porous medium basis. This is done by using the concepts of volume porosity and directional surface porosity. The volume porosity, γ_v , is defined as the fraction of the geometric grid volume of a computational cell which is occupied by fluid. The volume of fluid, V_{f0} , in computational cell (i,j,k) is, thus, given by

$$(V_{f0})_{ij,k} = (\gamma_v)_{ij,k} \Delta x_i \Delta y_j \Delta z_k \quad (3.3a)$$

in Cartesian coordinates and

$$(V_{f0})_{ij,k} = (\gamma_v)_{ij,k} \Delta r_i (r_i \Delta \theta_j) \Delta z_k \quad (3.3b)$$

in cylindrical coordinates, where r_i is the distance from the origin to the grid point in the center of the computational cell. In a similar manner, the directional surface porosities, γ_x or γ_r , γ_y or γ_θ , and γ_z , are defined as the fraction of the given direction's cell face area which is open to fluid flow and conduction heat transfer through the fluid. The various flow areas on the plus

Table 3.9. Boundary Surface Identification Input Block Record Structure

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier; may be 'IREG': an irregular surface is being defined; 'REG ': a regular surface is being defined; 'END ': terminates this input block; record positions 5-42 should be blank.	(see below)
5-14	F10.4	AREA	flow area [m ²] of each surface element. If AREA < 0: area will be set internally to geometric grid area (i.e., $\Delta x \Delta y$, $\Delta y \Delta z$, or $\Delta x \Delta z$ for Cartesian coordinates and equivalent values for cylindrical coordinates).	
15-18	I4	IB	beginning I-range index.	$1 \leq IB \leq IE$
19-22	I4	IE	ending I-range index.	$IB \leq IE \leq IMAX$
23-26	I4	JB	beginning J-range index.	$1 \leq JB \leq JE$
27-30	I4	JE	ending J-range index.	$JB \leq JE \leq JMAX$
31-34	I4	KB	beginning K-range index.	$1 \leq KB \leq KE$
35-38	I4	KE	ending K-range index.	$KB \leq KE \leq KMAX$
39-42	I4	N	surface number.	$1 \leq N \leq NSURF$
43-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. A Boundary surface is defined by the cell grid point indices (I,J,K) in the three-dimensional region delimited by $IB \leq I \leq IE$, $JB \leq J \leq JE$, and $KB \leq K \leq KE$ within the system volume which are (a) adjacent to the boundary surface and (b) toward which the surface-normal vector points.
2. The same surface number N may be used for all surfaces which have the same (a) unit surface-normal vector, (b) velocity, energy, and pressure boundary conditions, and (c) fluid number in the cells adjacent to the boundary surface.
3. Use as many records as needed to define each boundary surface. Records defining boundary surface $N=n$ must be grouped together, but their ordering is irrelevant. A given set of cell indices can only appear once for a given boundary surface number.
4. Records defining boundary surface $N=n+1$ immediately follow records defining boundary surface $N=n$.
5. All 'IREG' records must precede any 'REG' records.
6. The 'END' record only occurs once and must be last.

side of a cell at spatial location (i,j,k) are given by

$$(A_x)_{i+1/2,j,k} = (\gamma_x)_{i+1/2,j,k} \Delta y_j \Delta z_k \quad (3.4a)$$

$$(A_y)_{i,j+1/2,k} = (\gamma_y)_{i,j+1/2,k} \Delta x_i \Delta z_k \quad (3.5a)$$

$$(A_z)_{i,j,k+1/2} = (\gamma_z)_{i,j,k+1/2} \Delta x_i \Delta y_j \quad (3.6a)$$

in Cartesian coordinates and

$$(A_r)_{i+1/2,j,k} = (\gamma_r)_{i+1/2,j,k} (r_{i+1/2} \Delta \theta_j) \Delta z_k \quad (3.4b)$$

$$(A_\theta)_{i,j+1/2,k} = (\gamma_\theta)_{i,j+1/2,k} \Delta r_i \Delta z_k \quad (3.5b)$$

$$(A_z)_{i,j,k+1/2} = (\gamma_z)_{i,j,k+1/2} \Delta r_i (r_i \Delta \theta_j) \quad (3.6b)$$

in cylindrical coordinates, where $r_{i+1/2}$ is the distance from the origin to the boundary which separates computational cell (i,j,k) from computational cell (i+1,j,k).

The volume and directional surface porosities have a default value of 1, which is appropriate for calculations in continuum fields having regular geometry. The user may need to input porosities for irregular geometries and noncontinuum flow fields. For the examples in Fig. 3.7, consider what is shown to be a single computational cell surrounding grid point (i,j,k). For the irregular geometry shown in Fig. 3.7a, the user would need to input

$$(\gamma_v)_{i,j,k} = \frac{\Delta x_i \Delta y_j \Delta z_k - \text{volume of prism A}}{\Delta x_i \Delta y_j \Delta z_k} \quad (3.7a)$$

$$(\gamma_x)_{i+1/2,j,k} = \frac{\Delta y_j \Delta z_k - \text{area A1}}{\Delta y_j \Delta z_k} \quad (3.7b)$$

$$(\gamma_y)_{i,j+1/2,k} = \frac{\Delta x_i \Delta z_k - \text{area A2}}{\Delta x_i \Delta z_k} \quad (3.7c)$$

Fig. 3.7 Alteration of Volume and Surface Porosities by (a) Irregular Geometry and (b) Internal Solid

$$(\gamma_z)_{i,j,k+1/2} = \frac{\Delta x_i \Delta y_j - \text{area A3}}{\Delta x_i \Delta y_j} \quad (3.7d)$$

Since surface porosities are associated with the plus-side face of a computational cell, the modification of the y-direction porosity on the minus side of cell (i,j,k) is accomplished by changing the porosity on the plus side of cell (i,j-1,k). For the noncontinuum geometry shown in Fig. 3.7b, the user would need to input

$$(\gamma_v)_{i,j,k} = \frac{\Delta x_i \Delta y_j \Delta z_k - \text{volume of half solid cylinder}}{\Delta x_i \Delta y_j \Delta z_k} \quad (3.8a)$$

$$(\gamma_x)_{i+1/2,j,k} = \frac{\Delta y_j \Delta z_k - \text{area A5}}{\Delta y_j \Delta z_k} \quad (3.8b)$$

$$(\gamma_z)_{i,j,k-1/2} = (\gamma_z)_{i,j,k+1/2} = \frac{\Delta x_i \Delta y_j - \text{area A4}}{\Delta x_i \Delta y_j} \quad (3.8c)$$

The default value of γ_y has not been overridden since the cylinder does not fall on either the plus or minus y-direction face of cell (i,j,k); the default value of γ_z has been overridden on both the plus and minus faces of cell (i,j,k) since the cylinder has been assumed to occupy the full z-direction length of the cell.

The user must input the volume porosity for every computational cell which is adjacent to an irregular surface. The areas of boundary surface elements (1) on irregular surfaces and (2) on regular surfaces which are intersected by irregular surfaces are input using the "AREA" variable in the records in the Boundary Surface Identification input block rather than by inputting the directional surface porosity (which is not used on boundary surfaces); cell-to-cell flow areas affected by irregular boundary surfaces are input using directional surface porosities. A surface porosity of 0 may be input to denote a complete flow blockage between two computational cells; this method may also be used to separate adjacent cells which contain different fluid numbers, although this separation should normally be accomplished using coincident pairs of boundary

surfaces. If one of the fluid turbulence models corresponding to $ITURKE \geq 10$ (cf. Sec. 8) is to be used, a complete flow blockage between two computational cells must be represented by a coincident pair of boundary surfaces (having unit surface-normal vectors pointing in opposite directions), rather than by using a zero surface porosity. The volume porosity must always be strictly positive. If a region of the grid is to be totally void of fluid, it should be surrounded by a series of boundary surfaces which point toward the fluid in the rest of the system. With one exception, volume and surface porosities are always less than or equal to one; the single exception is the expansion cell associated with the $KFLOW = -6$ velocity boundary condition (cf. Sec. 4.1.1) for which $\gamma_v > 1$ is allowed.

The nondefault values of volume and surface porosity are entered by the user in the Internal Cell Initialization input block. This block consists of a set of fixed format records having the record structure shown in Table 3.10. As these records are read, the indicated arrays are filled with "VALUE" to override the default value of 1.0 for the indicated cells in the region of the system volume delimited by indices IB, IE, JB, JE, KB, and KE. The input values for volume porosity affect the volume and, thus, the mass of fluid in a computational cell and the input values for the surface porosities affect the areas for fluid flow and conduction heat transfer in the fluid. If a solid object such as that depicted in Fig. 3.7b is to exert a frictional force on the fluid or to exchange energy with the fluid, then additional user input is required as discussed in the Force Structure Model (Sec. 6) and the Thermal Structure Model (Sec. 5).

Table 3.10. Geometry Variables in Internal Cell Initialization Input Block

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier; may be 'AL ': "VALUE" is volume porosity, γ_v , [-]; 'ALX ': "VALUE" is x- or r-direction surface porosity, γ_x or γ_r , [-]; 'ALY ': "VALUE" is y- or θ -direction surface porosity, γ_y or γ_θ ; 'ALZ ': "VALUE" is z-direction surface porosity, γ_z , [-]; 'END ': terminates this input block; record positions 5-38 should be blank.	(see below)
5-14	F10.4	VALUE	see NAME.	$0 < AL \leq 1$ $0 \leq ALX \leq 1$ $0 \leq ALY \leq 1$ $0 \leq ALZ \leq 1$
15-18	I4	IB	beginning I-range index.	$1 \leq IB \leq IE$
19-22	I4	IE	ending I-range index.	$IB \leq IE \leq IMAX$
23-26	I4	JB	beginning J-range index.	$1 \leq JB \leq JE$
27-30	I4	JE	ending J-range index.	$JB \leq JE \leq JMAX$
31-34	I4	KB	beginning K-range index.	$1 \leq KB \leq KE$
35-38	I4	KE	ending K-range index.	$KB \leq KE \leq KMAX$
39-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. These records result in assigning the value "VALUE" to the variable named "NAME" for the spatial positions (I,J,K) in the three-dimensional region delimited by $IB \leq I \leq IE$, $JB \leq J \leq JE$, and $KB \leq K \leq KE$ within the system volume.
2. Use as many 'AL ', 'ALX ', 'ALY ', and 'ALZ ' records as needed to define volume and surface porosities; the order of the records is irrelevant, except if a cell or cell face is assigned a value in more than one record, the final such specification will be used.
3. For 'AL ', the indices (I,J,K) refer to cell-center grid points; the cells must actually exist (i.e. contain fluid).
4. For 'ALX ' the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I+1,J,K).
5. For 'ALY ', the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I,J+1,K).
6. For 'ALZ ', the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I,J,K+1).
7. For 'ALX ', 'ALY ', and 'ALZ ', the cell faces referenced must exist and must not lie on boundary surfaces.
8. These 'AL ', 'ALX ', 'ALY ', and 'ALZ ' record may be interspersed among the other records (i.e. 'P ', 'QSOU', 'TL ', 'TURC', 'TURV', 'UL ', 'VL ', and 'WL ') in the Internal Cell Initialization Input Block (cf. Tables 4.10 and 8.3). An 'END ' record terminates the input block, occurs only once, and must be last.

4. BOUNDARY AND INITIAL CONDITIONS

Boundary conditions must be specified in order to solve any system of differential equations. Initial conditions must also be specified for equations to be solved in a time dependent manner. This section describes the boundary and initial conditions available in COMMIX for the fluid mass, momentum, and energy conservation equations. The similar conditions for the structure energy equations, the pump equations, and the fluid turbulence equations will be presented along with other input for these models in Secs. 5, 7, and 8, respectively, since these models are optional and their inputs are not in the format discussed here.

4.1 Boundary Conditions

A physical fluid system to be modeled in COMMIX must have the collection of fluid cells completely enclosed by a set of rigid nonmoving walls as described in Sec. 3. (There is no provision for modeling a true free surface.) These walls may have openings which provide paths through which fluid can enter and leave the volume. The impervious parts of the walls may be energy transfer paths. If symmetry arguments are invoked to avoid modeling the full geometry, then the cuts performed to produce a sector geometry are symmetry boundaries. Each of these walls, openings, and symmetry cuts is identified as a "boundary surface".

Each boundary surface may have a different combination of boundary conditions; however, all elements of a single boundary surface must have the same combination of velocity, pressure, and energy boundary conditions. This means that a given physical surface may need to be divided into several boundary surfaces in using COMMIX. (Example: A wall with an opening must be divided into at least two boundary surfaces: one for the opening and one (or more) for the impervious part.) Some physically based rules govern how the various boundary condition types may be used to surround the fluid. (Example: if one boundary surface is an inlet, then (at least) one boundary surface must be an outlet.) Care must be taken that the combination of specified boundary conditions is sufficient (in a numerical sense) for a unique solution to be obtained. (Example: Temperature can not be uniquely determined if all energy boundary conditions are of the heat flux type.) The structure of COMMIX also places some restrictions

on how boundary conditions may be combined. (Example: A specified-pressure-type pressure boundary condition may only be used on boundary surfaces which also use a continuative-type velocity boundary condition.)

Three types of boundary conditions must be specified for each boundary surface: velocity, pressure, and energy. The options available for each type of boundary condition are discussed in the next three subsections; this discussion parallels that of Sec. 10.1 of the Equations and Numerics report (Ref. 1). The user input for boundary conditions consists of an integer array for each type of boundary condition (KFLOW, KPRES, and KTEMP) which selects the option and a floating-point array (VELOC, PRES, and TEMP) which specifies a reference value needed for some options; several energy boundary condition options require input in other arrays. Some boundary conditions use "transient functions" to prescribe a known variation; the input associated with transient functions is described in Sec. 4.1.4. All user input associated with boundary conditions will be summarized in Sec. 4.1.5 by input block type.

4.1.1 Velocity

The options available for velocity boundary conditions fall into three broad categories: specified value, continuative, and free slip. In all cases, what is being specified (either directly in value or indirectly by a relational formula) is the "boundary normal velocity", u_n , which is the velocity at the boundary surface in the direction of the unit surface-normal vector. The sign convention for the boundary normal velocity is such that $u_n > 0$ denotes flow into the system volume (just as the unit surface-normal vectors point from the boundary toward the fluid). Table 4.1 shows the various options available, associating these with the various physical types of boundaries.

Table 4.1 Types of Velocity Boundary Conditions

Surface Type	Suitable Option	Input Arrays		Remarks
		KFLOW	VELOC ¹	
Inlet	specified velocity:			
	constant	1	u_{w0}	
	transient function:			
	of time	100+nf	u_{w0}	See note 2.
	of radiation surface temperature	200+nf	u_{w0}	See note 2.
	continuative mass flow	-5	n.a.	Use when pressure is to be specified.
Outlet	continuative:			
	mass flow	-5	n.a.	Most general.
	mass flux	-1	n.a.	Suitable for constant area.
	velocity	-2	n.a.	Suitable for constant area and density.
	uniform velocity	-4	n.a.	Rarely used.
Impervious	zero velocity	1	0	Normal wall; $\tau_w \neq 0$.
	free slip	-3	n.a.	$\tau_w = 0$
Symmetry	free slip	-3	n.a.	$\tau_w = 0$.
Miscellaneous	expansion cell	-6	n.a.	Constant mass system should have this on one surface.
Irregular	zero velocity	1	0	$\tau_w \neq 0$.
	free slip	-3	n.a.	$\tau_w = 0$.

Notes:

1. The notation "n.a." means not applicable; any value input will be ignored.
2. User must define transient function number nf ($1 \leq nf \leq 25$).

At an inlet surface, either the velocity or the pressure must be specified. If the velocity is known, one of the specified-value options should be selected. The velocity may be constant in time (selected by KFLOW=1)

$$u_n = u_{n0} , \quad (4.1)$$

a known function of time (KFLOW=100+nf)

$$u_n = u_{n0} f_{nf}(t) \quad (4.2)$$

or a known function of the temperature of a specific radiation heat transfer surface (KFLOW=200+nf)

$$u_n = u_{n0} f_{nf}(T_{r,nf}), \quad (4.3)$$

where u_{n0} is a reference value, f_{nf} is transient function number nf, t is time, and $T_{r,nf}$ is the temperature of radiation surface number nf. (Radiation surfaces are discussed in Sec. 5.) The user must input the reference value and the transient function. For these specified-value options, the value calculated from Eq. 4.1, 4.2, or 4.3 is assigned uniformly to each surface element which forms the boundary surface. If the pressure is known at the inlet, the continuative-mass-flow velocity boundary condition (described in the next paragraph) should be used at the inlet.

An outlet surface is described using one of the continuative velocity boundary conditions. All four of the available options relate the velocity on the boundary to the velocity across the cell face which is one mesh spacing into the fluid from the boundary. The most general of the four options is the continuative-mass-flow velocity boundary condition (KFLOW=-5). Using the nomenclature of Fig. 4.1, the boundary velocity will be calculated using

$$u_{n,\ell-1/2} = \frac{(A_f u_i)_{\ell+1/2} \rho_{\ell+1}^i + \left(V_f \frac{d\rho}{dt} \right)_i}{(A_f \rho)_{\ell-1/2}} \quad (4.4a)$$

for the boundary surface associated with cell ℓ and

Fig. 4.1 Cells Near Outlet Boundary Surfaces

$$u_{n,m+1/2} = \frac{-(A_f u_i)_{m-1/2} \langle \rho \rangle_m^{m-1} + \left(V_f \frac{d\rho}{dt} \right)_m}{(A_f \rho)_{m+1/2}} \quad (4.4b)$$

for the boundary surface associated with cell m , where ρ is the fluid density, A_f is the fluid flow area for a cell face, V_f is the fluid volume of a cell, and u_i is the fluid velocity in coordinate direction i which is the same direction as the unit surface-normal vector. The density, $\langle \rho \rangle$, appearing in the numerator of these equations is for the upwind cell based on the sign of the velocity at the indicated cell face. The density in the denominator is associated with the boundary surface and is defined in terms of the boundary enthalpy and pressure. The arrangement of signs in these equations is different due to the normal velocity (u_n) being defined as positive for flow into the fluid domain. The condition given by Eq. 4.4 is applied individually to each cell face which is part of the boundary surface. The continuity balance given by Eq. 4.4 essentially forces there to also be, on a cell-wise basis for each of the cells on

the boundary surface, zero net mass flow for the sum of the two directions which are transverse to the unit surface-normal vector. In some cases where there are multiple surface elements forming a boundary surface, the user must place "flow straighteners" adjacent to the boundary surfaces using continuative velocity boundary conditions in order to force there to be zero net transverse flow; this is accomplished by inputting a zero surface porosity for the cell faces which are parallel to the unit surface-normal vector (using 'ALX ', 'ALY ', and/or 'ALZ ' records in the Internal Cell Initialization input block as described in Sec. 3.5). The areas A_f must not be totally blocked to fluid flow.

The other continuative options are based on the continuative-mass-flow option with additional assumptions imposed. The continuative-mass-flux option (KFLOW=-1) assumes that the two flow areas appearing in Eq. 4.4 are identical. The continuative-velocity option (KFLOW=-2) assumes that the two areas are identical and the two densities are identical in Eq. 4.4. These two options should only be selected by the user if the area and density restrictions are truly satisfied; a safer approach is to always use the continuative-mass-flow boundary condition. The fourth continuative boundary condition, uniform velocity (KFLOW=-4), is implemented by replacing both the numerator and the denominator in Eq. 4.4 with summations over all surface elements on the boundary surface; the resulting average velocity is then assigned to all elements of the boundary surface. The uniform-continuative-velocity option is seldom used; it will produce nonphysical results if the surface to which it is applied does not truly have a uniform velocity in the real physical system being modeled.

An impervious surface should be modeled using the specified-constant-velocity option with the reference value, u_{n0} , being zero. This precludes fluid flow across the boundary surface. This also imposes a zero-slip boundary condition for flow in the directions which are transverse to the unit surface-normal vector at the boundary surface; the wall shear, τ_w , is, thus, not zero. This zero-velocity boundary condition must be selected in order to activate the wall function treatment of shear stress which is a part of the nonconstant turbulence models (cf. Sec. 8). If zero wall shear is desired, then the free-slip option (KFLOW=-3) should be selected for the surface.

A surface forming a symmetry boundary should be modeled using the free-slip option. This precludes fluid flow across the boundary surface; however, unlike the above zero-velocity option for impervious surfaces, there is no wall shear imposed by the free-slip option. In cylindrical coordinates, the boundary surface at $r=0$ should be assigned a free-slip velocity boundary condition.

A final type of velocity boundary is denoted "expansion cell". A region of fluid cells should not usually be totally surrounded by boundary surfaces having zero-velocity and free-slip velocity boundary conditions; this arrangement provides no allowance for fluid expansion and contraction and will cause severe numerical problems for an equation of state where small changes in density lead to large changes in pressure (e.g. a liquid). Instead, one extra computational cell, denoted the expansion cell, should be added near the top of the normal system geometry as shown in Fig. 4.2. This cell is open to fluid flow at the point of attachment; the cell face on the opposite side is assigned an expansion-cell velocity boundary condition ($KFLOW=-6$); the other four cell faces must be impervious to fluid flow (and heat transfer). No thermal structures should interact with this cell. During a transient calculation, the expansion-cell velocity boundary condition acts exactly like the continuative-mass-flow velocity boundary condition until the time step converges. After convergence, the mass which appeared to leave (enter) the system across cell boundary b is added to (removed from) the volume of fluid in expansion cell m . This is accomplished by modifying the volume porosity of the expansion cell:

$$\gamma_{v,m}^n = \gamma_{v,m}^{n-1} - \frac{(\rho A_f u_n)_b (t^n - t^{n-1})}{(\rho V)_m} \quad (4.5)$$

where the subscript b refers to the boundary surface element (either $m-1/2$ or $m+1/2$ depending on whether the boundary surface is on the minus or plus side of the expansion cell) and V is given by

$$V = \Delta x \Delta y \Delta z \quad (4.6a)$$

Fig. 4.2 Example Placement of an Expansion Cell

in Cartesian coordinates and

$$V = \Delta r \, r \Delta \theta \, \Delta z \quad (4.6b)$$

in cylindrical coordinates. The volume (and thus mass) of fluid in the expansion cell increases (decreases) as the fluid in the rest of the system expands (contracts); the total system mass remains constant. (During a steady-state calculation, this boundary condition behaves exactly like a continuative-mass-flow velocity boundary condition.) Although the volume porosity of the expansion cell changes during the calculation and is allowed to exceed one, it must never go negative; for this reason the initial volume of fluid in the expansion cell must be sufficiently big to provide a source of enough fluid into the rest of the system for the largest amount of contraction anticipated during the transient. This initial sizing is set by the product of the grid volume V (which is related to the grid coordinates by Eq. 4.6) and the initial value specified for the volume porosity γ_v (which is entered via an 'AL' record in the Internal Cell Initialization

input block as described in Sec. 3.5). Several expansion cells may be needed in a calculation, each attached to a different fluid region; there is a limit of 10 expansion cells in a calculation.

The preceding options are all available for regular boundary surfaces. Irregular boundary surfaces may only use the zero-velocity and free-slip options for velocity boundary conditions.

Some rules govern how the various velocity boundary conditions may be used to surround a fluid system. If there is one (or more) inlet boundary surface(s), then there must be (at least) one outlet boundary surface. Similarly, if there is one (or more) outlet boundary surface(s), then there must be (at least) one inlet boundary surface. Specified-velocity boundary conditions should only be used at inlet boundary surfaces. Either velocity or pressure, but not both, may be specified as an inlet boundary condition. A region of fluid cells should not usually be totally surrounded by zero-velocity and free-slip velocity boundary conditions; instead, add an expansion cell having an expansion-cell velocity boundary condition on one of its surfaces. The boundary condition given by Eq. 4.3 is particularly well suited for modeling the passive cooling of a nuclear reactor vessel exterior by natural circulation of air; the air flow rate may usually be correlated as a function of the reactor vessel wall temperature for a specific geometrical arrangement of the various components, obviating the need to model the entire flow path of the air; note that "nf" is used as both the transient function number and the radiation surface number.

4.1.2 Pressure

The options available for pressure boundary conditions fall into two broad categories: specified value and not specified. Table 4.2 shows the various options available, associating these with the various physical types of boundaries.

At an outlet, one of the specified-value pressure boundary conditions should normally be used (in conjunction with a continuative velocity boundary condition). The specified pressure may be constant (selected by KPRES=1)

$$P_b = P_{b0}, \quad (4.7)$$

Table 4.2 Types of Pressure Boundary Conditions

Surface Type	Suitable Option	Input Arrays		Remarks
		KPRES	PRES ¹	
Inlet	not specified	0	n.a.	Most common; requires KFLOW > 0.
	specified pressure:			Allowed only if KFLOW < 0.
	constant	1	P _{b0}	
	function of time	100+nf	P _{b0}	See note 2.
Outlet	specified pressure:			Most common; requires KFLOW < 0.
	constant	1	P _{b0}	
	function of time	100+nf	P _{b0}	See note 2.
	not specified	0	n.a.	See text.
Impervious	not specified	0	n.a.	Required.
Symmetry	not specified	0	n.a.	Required.
Irregular	not specified	0	n.a.	Required.
Expansion Cell	See "Outlet".			For surface with KFLOW = -6.

Notes:

1. The notation "n.a." means not applicable; any value input will be ignored.
2. User must define transient function number nf ($1 \leq nf \leq 25$).

or a function of time (KPRES = 100 + nf)

$$P_b = P_{b0} f_{nf}(t), \quad (4.8)$$

where P_{b0} is a reference value and f_{nf} is transient function number nf. The user must input the reference value and the transient function. For these specified-value options, the value calculated from Eq. 4.7 or 4.8 is assigned uniformly to each surface element which forms the boundary surface. This same value is also assigned as the pressure at the centers of the

computational cells which are adjacent to the boundary surface and toward which the unit surface-normal vector points; the pressure in each of these cells, thus, become a "known" rather than an "unknown" in the finite-difference formulation. For this reason, the specified-value option should only be used for a boundary surface which either has (a) a single boundary surface element or (b) multiple surface elements but a unit surface-normal vector which is parallel to the gravity vector and which is expected to have flow parallel to the unit surface-normal vector; examples of these geometries are shown in Fig. 4.3. If the not-specified pressure boundary condition ($KPRES=0$) is used at a surface having a continuative velocity boundary condition, then the boundary pressure (and the pressure in the adjacent cells) remains at the initial value (cf. Sec. 4.2).

Fig. 4.3 Recommended Boundary Surface Arrangements for Pressure Boundary Conditions

At an inlet surface, either the pressure or the velocity must be specified. If the pressure is known, one of the specified-value pressure boundary conditions described above should be used in conjunction with a continuative velocity boundary condition. If the velocity is known, then the not-specified option must be selected for the pressure boundary condition; the pressure at the

inlet surface (and in the adjacent cells) will be determined from the pressure boundary conditions on the other surfaces and the resulting flow field.

The not-specified pressure boundary condition must always be used for impervious, free-slip, and irregular surfaces. The pressures on these surfaces (and in the adjacent cells) will be determined from the pressure boundary conditions on the other surfaces and the resulting flow field.

The surface of an expansion cell which is assigned the expansion-cell velocity boundary condition should be viewed as an "outlet" in choosing the pressure boundary condition. The other four boundary surfaces of the expansion cell must be assigned the not-specified pressure boundary condition. Each one of the five boundary surfaces of an expansion cell has only a single surface element.

Some rules govern how the various pressure boundary conditions may be used. The specified-value options may only be used on surfaces having a continuative velocity boundary condition (which should normally be continuative-mass-flow). The result of the not-specified option depends on the type of velocity boundary condition: surface (and adjacent cell) pressures (a) stay at their initial values on surfaces having continuative velocity boundary conditions or (b) are determined by other surfaces' pressure boundary conditions and the resulting flow field on surfaces having other velocity boundary conditions. Pressure must be specified on at least one boundary surface for each fluid region in order for the pressure field to be uniquely determined (in a mathematical sense). Pressures in COMMIX include a gravity head effect if the user-input gravitational acceleration vector is nonzero (cf. Sec. 3.2).

4.1.3 Energy

The options available for energy boundary conditions fall into two broad categories: specified temperature or specified heat flux. For both categories, the value is either specified directly or a method is prescribed for indirect calculation of the value. The sign convention is such that heat transfer from the boundary to the fluid is positive. Throughout this section, the

heat flux at the boundary, q_b , is the amount of heat transferred per unit area in absence of any mass flow; if there is mass flow across the boundary surface, this results in an additional flow rate of energy into or out of the fluid system, which has the form (mass flow rate)*(fluid enthalpy). Table 4.3 shows the various options available, associating these with the various physical types of boundaries.

At an inlet surface, the temperature on the boundary, T_b , must be specified. The value may be constant (selected by KTEMP=1)

$$T_b = T_{b0} , \quad (4.9)$$

a function of time (KTEMP=100+nf)

$$T_b = T_{b0} f_{nf}(t), \quad (4.10)$$

or the current value of temperature on outlet boundary surface m (KTEMP=600+m)

$$T_b = \frac{\sum_{i \in m} A_{b,i} T_{b,i}}{\sum_{i \in m} A_{b,i}}, \quad (4.11)$$

where T_{b0} is a reference value and f_{nf} is transient function number nf. The user must input the reference value and the transient function. The condition given by Eq. 4.11 is denoted a recirculation-temperature energy boundary condition; the summations are over all surface elements which form boundary surface m. For these specified-value options, the value calculated from Eq. 4.9, 4.10, or 4.11 is assigned uniformly to all surface elements which form the boundary surface. The boundary temperature given by Eq. 4.9 or 4.10 is used in conjunction with the fluid temperature, T_f , in the computational cell adjacent to the boundary surface element and pointed to by the unit surface-normal vector to calculate a heat flux, q_b , from the boundary to the fluid using

$$q_b = U_b (T_b - T_f), \quad (4.12)$$

Table 4.3 Types of Energy Boundary Conditions

Surface Type	Suitable Option	Input Arrays		Remarks
		KTEMP	TEMP ¹	
Inlet	specified temperature:			
	constant	1	T_{bo}	
	function of time	100+nf	T_{bo}	See note 2.
	recirculation, using temperature of surface m	600+m	n.a.	Surface m must be an outlet.
	adiabatic	400	n.a.	ISTATE ≥ 2; KPRES > 0
Outlet	adiabatic	400	n.a.	
Impervious	specified temperature:			
	constant	1	T_{bo}	
	function of time	100+nf	T_{bo}	See note 2.
	specified heat flux:			
	adiabatic	400	n.a.	
	constant	200	q_{bo}	
	function of time	300+nf	q_{bo}	See note 2.
	duct wall	500+nf	n.a.	See notes 2 & 3.
Symmetry	adiabatic	400	n.a.	Required.
Irregular	See "Impervious".			
Expansion Cell	adiabatic	400	n.a.	All 5 surfaces.

Notes:

1. The notation "n.a." means not applicable; any value input will be ignored.
2. User must define transient function number nf ($1 \leq nf \leq 25$).
3. Additional input is required in arrays IHTWAL, HYDWAL, MATWAL, WALLDX, TSINK, HSINK, WALLQS, QIN, and QK; transient function is not used if nf=0.
4. For KTEMP < 400; additional input may be entered in arrays IHTWAL, HYDWAL, MATWAL, and WALLDX.

where U_b is a heat transfer coefficient given by

$$U_b = \left[\frac{1}{U_f} + \left(\frac{L/2}{\lambda_w} \right) \right]^{-1}. \quad (4.13)$$

The quantity U_f will be computed in one of two ways. (1) If the user enters a nonzero value in array IHTWAL, this value is the index number of one of the user-defined heat transfer correlations (cf. Sec. 9.3); U_f will be computed using this correlation and the local fluid conditions; the corresponding characteristic length must be entered in array HYDWAL. (2) If IHTWAL=0, U_f will be computed by the code based on conduction through a half thickness of the fluid cell using

$$U_f = \frac{\lambda_f}{0.5 \frac{V_f}{A_b}}, \quad (4.14)$$

where λ_f is the total (laminar plus turbulent) thermal conductivity of the fluid, V_f is the fluid volume in the cell, and A_b is the flow area of the boundary surface element. In a similar manner, the second term in Eq. 4.13 may be computed in one of two ways. (1) If the user enters a nonzero value in array MATWAL, this value is the index number of one of the structure properties correlation sets defined by the user (cf. Sec. 9.2); the thermal conductivity, λ_w , will be evaluated at the boundary temperature using this material's correlation; the structure thickness, L_w , must be input in array WALLDX. (2) If MATWAL=0, the solid conduction term in Eq. 4.13 is neglected. An inlet surface would normally have IHTWAL=0 and MATWAL=0. When the inlet surface temperature is calculated from Eq. 4.11, the corresponding boundary heat flux is zero, rather than being calculated from Eq. 4.12.

An inlet boundary surface should not normally use the adiabatic energy boundary condition (KTEMP=400). The only reason for including it in the list of options is for transient cases which have a specified-pressure boundary condition at the inlet for which reversal of the flow field is expected (i.e. the inlet becomes an outlet). The specified-temperature boundary condition at the inlet would cause problems following flow reversal. The adiabatic boundary

condition avoids these problems; however, this is not the normal purpose for which the adiabatic energy boundary condition was designed (which is discussed in the next paragraph).

An outlet surface should be described using the adiabatic energy boundary condition. The boundary heat flux, q_b , will be zero. The boundary temperature, T_b , will be set to the temperature of the fluid in the computational cell adjacent to the boundary and pointed to by the unit surface-normal vector for each boundary surface element.

An impervious surface has the broadest range of options available. The boundary temperature may be specified as constant or as a function of time, using equations 4.9 and 4.10, respectively, and having the same options and ramifications as discussed for inlet surfaces. The heat flux may be specified, instead of temperature, as a constant (KTEMP=200)

$$q_b = q_{b0} \quad (4.15)$$

or as a function of time (KTEMP=300+nf)

$$q_b = q_{b0} f_{nf}(t), \quad (4.16)$$

where q_{b0} is a reference value and f_{nf} is transient function number nf. The user must input the reference value and the transient function. The heat flux given by Eq. 4.15 or 4.16 is assigned uniformly to all surface elements which form the boundary surface. The boundary heat flux is used in conjunction with the fluid temperature to calculate a boundary temperature, using a rearrangement of Eq. 4.12

$$T_b = T_f + q_b / U_b, \quad (4.17)$$

where U_b is given by Eq. 4.13 with the options as discussed above for IHTWAL and MATWAL. An impervious surface may also use the adiabatic energy boundary condition, as described above for an outlet surface.

The final energy boundary condition option for an impervious surface is a duct wall (KTEMP=500+nf). This boundary condition allows a finite-thickness wall to be placed between a constant sink temperature and the fluid. Each boundary surface element has its own

wall segment and is treated separately using a lumped-heat-capacity method. Using the geometry shown in Fig. 4.4, the energy balance for a wall segment is given by

$$(\rho C_p L)_w \frac{dT_b}{dt} = U_b (T_f - T_b) - U_s (T_b - T_s) + \dot{q}''_w L_w, \quad (4.18)$$

where ρ_w and C_{pw} are the density and specific heat capacity of the wall material and are computed from the structure properties correlation set for the material specified by the entry in array MATWAL, L_w is the wall thickness specified in array WALLDX, and T_s is the (constant) temperature for the surroundings on the nonfluid side of the wall specified in array TSINK. The heat transfer coefficient on the fluid side, U_b , comes from Eq. 4.13 and the input in arrays IHTWAL, HYDWAL, MATWAL, and WALLDX. The coefficient on the other side, U_s , is constant and is input in array HSINK. The heat source in the wall, if used, is assumed to be in the following separable form

$$\dot{q}''_w = QJ(i,j) * QK(k) * \dot{q}''_{w0} * f_{nf}(t), \quad (4.19)$$

where $QJ(i,j)$ is the normalized transverse power distribution at location (i,j) for all axial locations k and is input using array QIN ($QJ(i,j) = QIN(i + IMAX*(j-1))$), $QK(k)$ is the input array giving the normalized axial power distribution at axial location k , $\dot{q}''_{w0}$ is a reference volumetric heat generation rate input in array WALLQS, and f_{nf} is transient function number nf . Equation 4.18 is integrated in time by the code during each time step to obtain the new boundary surface temperature. The boundary heat flux is then calculated using Eq. 4.12.

A symmetry surface must use the adiabatic option to preserve the physical situation imposed by symmetry. All five surfaces of an expansion cell must use the adiabatic option to be consistent with the formulation in the code. An irregular surface may use any of the options described for an impervious surface.

Some rules govern how the various energy boundary conditions may be used to surround a fluid system. A specified-temperature or the duct-wall option must be used on at least one boundary surface in order for the temperature field to be determined uniquely (in a numerical

Fig. 4.4 Finite-Thickness Wall Boundary Surface

sense). Although a zero-heat flux condition may be imposed using either the adiabatic option or the specified-heat-flux option with $q_{b0}=0$, the adiabatic option is the normal choice and will execute somewhat faster. When separating two fluid regions with a pair of coincident boundary surfaces (having unit surface-normal vectors pointing in opposite directions), both of these surfaces should normally have adiabatic energy boundary conditions; heat transfer from one fluid region to the other may be accomplished by using the thermal structure model (cf. Sec. 5).

One additional energy "boundary" condition is available which allows the specification of internal heat generation directly within the fluid. This heat source, on a per unit volume of fluid basis, is q''' in the source term for the energy equation (cf. Table 2.1 of Ref. 4.1). This may be useful to simulate heating by nuclear radiation or to model other heat sources without using the thermal structure model where the value of the heat transfer rate from the structure to the fluid is known. The form of this heat source term depends on whether thermal structures are being used. If there are no thermal structures in the system, the heat source must be specified as

$$q''' = \text{QSOUR}(i,j,k) * f_{\text{NOFQT}}(t), \quad (4.20a)$$

where $Q_{SOUR}(i,j,k)$ is a different reference value at each spatial position (i,j,k) and f_{NOFQT} is transient function NOFQT which is used for all spatial positions. If thermal structures are being used anywhere in the system volume, then the heat source must be specified as constant in time (but may be variable in space) using the form

$$q''' = Q_{IJ}(i,j) * Q_K(k) * Q_{SCOOL}, \quad (4.20b)$$

where Q_{SCOOL} is a reference value and Q_{IJ} and Q_K are the same shape functions described in connection with Eq. 4.19. The user must input the reference value(s), the shape functions, and the transient function. The internal heat generation rate in the fluid may be specified for any set of computational cells (within the restrictions imposed by the form of Eq. 4.20) and is not limited to just cells adjacent to boundary surfaces. A heat source which is inside of a submerged structure (e.g. a fuel pin) would normally be modeled using the thermal structure model (cf. Sec. 5) rather than approximated as an internal heat generation rate directly in the fluid.

4.1.4 Transient Functions

The concept of a transient function has been introduced in connection with several boundary conditions. This is the manner in which a boundary condition is specified as a known function of time. Transient functions are also associated with the internal heat generation directly in the fluid, heat sources in submerged structures (cf. Sec. 5 for the thermal structure model), the special form of the heat source for hexagonal fuel bundles (cf. Sec.10), and the several pump models (cf. Sec. 7). The functions are input by the user as described in this section.

Each transient function is identified by a number; these numbers are the subscript on " $f(t)$ " in the various equations and correspond to (a) the " nf " appearing in the entries for the boundary-condition-type identification arrays K_{FLOW} , K_{PRES} , and K_{TEMP} , (b) NOFQT used in internal heat generation in the fluid and in the hexagonal fuel bundle model, (c) NT used in the thermal structure heat source term, and (d) NTPUMP and NTORQM used in the pump models. Transient function numbers must be used contiguously beginning at 1. A maximum of 25 transient functions may be used in any calculation. The same transient function number

may be used for more than one purpose if that adequately describes the physical conditions in the system being modeled. For example, if the inlet velocity and the inlet temperature (at the same or different boundary surfaces) were both described by transient function 2, there would be an entry of "102" in the KFLOW and the KTEMP arrays for the appropriate boundary surfaces.

Transient functions are described by the user to the code using input arrays TVAL, FVAL, and NEND in NAMELIST DATA as shown in Table 4.4. This information must be provided at the beginning of a transient calculation (i.e. ISTATE=2). Each transient function is defined by a series of user-provided data points $(x_i, f(x_i))$, where the x_i values are entered in array TVAL and the corresponding function values are entered in array FVAL. Entries in array NEND(j) indicate the number of data points used in the TVAL and FVAL arrays to define transient function j.

Table 4.4 Transient Function Definition Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
TVAL(NP)	x : values of the independent variable, which is time [s] except for KTEMP=200+nf in which case it is temperature [°C], for transient functions.	$1 \leq NP \leq 100$
FVAL(NP)	$f(x)$: values of the dependent variable [-] for transient functions.	$1 \leq NP \leq 100$
NEND(NTF)	number of data points used to define transient function number NTF.	$1 \leq NTF \leq 25$ $\sum_{i=1}^{NTF} NEND(i) \leq 100$

Notes:

1. Arrays TVAL, FVAL, and NEND are used to define all transient functions used in the code by entering a series of data points $(x, f(x))$ to which the code will perform cubic-spline fits. The first NEND(1) entries in TVAL and FVAL are the data points for function number 1; the next NEND(2) entries in TVAL and FVAL are for function number 2; etc.
2. The entries in TVAL for each function must be nondecreasing. Entering the same value twice in succession in the TVAL array causes the derivative of the resultant fit to be discontinuous at that point; if the corresponding values in FVAL are different, the resultant fit will also be discontinuous in value.

The dependent variable, x_i , is always time, t , in unit of seconds except for the case $KTEMP=200+nf$ for which x_i is temperature in degrees Celsius. For each transient function, the values of the independent variable must be entered in nondecreasing order, $x_i \leq x_{i+1}$. These values should completely cover the expected range of the dependent variable during the duration of the transient calculation; if the code needs to evaluate a transient function outside of the defined range, the value of the function at the closest endpoint will be used -- no extrapolation is performed beyond these endpoints. In particular, a data point should be input corresponding to $t=0$ when x_i is time.

The data points defining transient function number $j+1$ immediately follow those defining transient function j in the TVAL and FVAL arrays; e.g. entries in location $NEND(1)+NEND(2)+1$ in the TVAL and FVAL arrays are the first data point used in defining transient function number 3. There may be no more than 100 entries in each of the TVAL and FVAL arrays; within this limitation, any number of data points may be used to define any given transient function. For example, if 10 transient functions are being used, the average number of data points used to describe each function would have to be 10 or fewer -- 5 functions could use 5 data points each and 5 functions could use 15 data points each. The number of entries in the NEND array corresponds to the number of functions being defined; the sum of the values entered in the NEND array must not exceed 100.

The data points entered in the TVAL and FVAL arrays are used by the code to generate a series of least-squares fits to describe each function, rather than using the points directly in a table-lookup mode. These resulting fits are third-order polynomials in the dependent variable and are performed between each pair of consecutive entries in the TVAL array. This results in each transient function being represented by a set of polynomials of the form

$$f_i(x) = a_i + b_i x + c_i x^2 + d_i x^3 \quad \text{for } x_i \leq x \leq x_{i+1}, \quad (4.21)$$

where the a , b , c , and d coefficients are the result of the least-squares fitting process. The fits are normally piecewise continuous in value and first derivative; i.e.

$$f_i(x) = f_{i+1}(x) \quad \text{at } x = x_{i+1} \quad (4.22)$$

and

$$\frac{df_i(x)}{dx} = \frac{df_{i+1}(x)}{dx} \quad \text{at } x = x_{i+1}. \quad (4.23)$$

If the same value is entered in two consecutive locations of both the TVAL and the FVAL arrays, the resulting polynomial functions will be discontinuous in derivative at this point. If the same value is entered in two consecutive locations in the TVAL array and the values in the corresponding locations in the FVAL array are not identical, the resulting polynomial functions will be discontinuous in value and derivative at this point; this is not a recommended practice; this type of step function should normally be spread out over at least a small range.

After the code has generated the polynomial fits, a table of values calculated from these fits will be printed by the code. The user is cautioned to check these tables carefully to insure that the polynomial fits do indeed give a proper rendering of the desired functional behavior. Generally, data points need to be concentrated in the area of rapid functional changes to get a proper behavior.

As an example, consider a case in which two transient functions are to be used in a calculation. Both functions are to be defined over a time range of 5 s. Transient function number 1 has a periodic variation over 4 s, passing through the data points (0,1), (1,1.3), (2,1), (3,.7), and (4,1), and is constant thereafter. The second transient function approximates a step function which is constant at a value of 1 for 1 s, then decreases to a value of 0.1 thereafter. The following user input (in NAMELIST notation):

```
TVAl=0.,1.,2.,3.,2*4.,5., 0.,2*1.,2*1.1,5.,
FVAL=1.,1.3,1.,.7,3*1., 3*1.,3*.1,
NEND=7, 6,
```

is recommended for defining these two functions. The first 7 values in the TVAL and FVAL arrays define transient function number 1; the point at 4 s is repeated to force a discontinuous slope at this point, in order for the function value to be constant over the remaining 1 s. The next 6 values in the TVAL and FVAL arrays define transient function number 2; the step change

in value is introduced over 0.1 s; the values at 1 and 1.1 s are both repeated, in order for the resulting fits to be three straight-line segments with no over- or under-shoots at either point. The functional behaviors resulting from the polynomial fitting process in the code for the above input data are shown in Fig. 4.5.

Fig. 4.5 Examples of Transient Functions

Normally, once the transient functions are defined at the beginning of a transient calculation (ISTATE=2), the input for TVAL, FVAL, and NEND should be removed from the input deck for subsequent runs which continue the calculation (ISTATE=3). The user may find situations in which it becomes necessary to change the definition of one (or more) transient functions during the course of a calculation; for example, the calculation needs to be continued to a time which exceeds the range over which the transient functions were originally defined. If this becomes necessary, the user may redefine the transient functions by including entries for the TVAL, FVAL, and NEND arrays in an ISTATE=3 continuation run. The user must, however, include data in these arrays to define all transient functions being used in the calculation, not just the ones to be changed. Additionally, the user must input a zero as the final entry in the NEND

array; for example, if 4 transient functions are being used and they are defined using 5, 10, 5, and 7 points, respectively, the input for the NEND array should be

NEND=5,10,5,7,0,

in order for the information in the TVAL and FVAL arrays to be properly processed. The new functions should be continuous in value with the old functions at the time in the calculational history at which the change is made in order to avoid convergence problems.

Transient functions are only intended for use during transient calculations ($ISTATE \geq 2$). Steady-state calculations ($ISTATE \leq 1$) should be performed using boundary conditions which do not involve the use of transient functions (e.g. a specified-value velocity boundary condition should be selected using $KFLOW=1$ rather than using $KFLOW=100+nf$) and using constant heat sources (i.e. $NOFQT=0$ for the direct fluid source and $NT=0$ in thermal structures). If transient functions are referenced in the input file at steady state, they should be defined as having the constant value of 1 for all time using input of the form

TVAL=0.,1.E20, 0.,1.E20, etc.,
FVAL=1.,1., 1.,1., etc.,
NEND=2, 2, etc.

4.1.5 User Input

The user inputs boundary conditions using variables in NAMELIST DATA and NAMELIST INPUTQ as shown in Tables 4.5, 4.6, and 4.7 at the beginning of a steady-state calculation ($ISTATE=0$). The primary inputs are the three integer arrays KFLOW, KPRES, and KTEMP shown in Table 4.5 which select the type of velocity, pressure, and energy boundary condition for each boundary surface. If the boundary condition is of the specified-value type, then the value is specified in one of the three companion floating-point arrays VELOC, PRES, and TEMP shown in Table 4.5; these values are assigned uniformly to all surface elements on the boundary surface. The values entered in VELOC and PRES are used to fill the proper locations in boundary surface element arrays VELBN and PB, respectively; for $KTEMP=1$ or $100+nf$,

Table 4.5 Boundary Condition Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
KFLOW(N)	<p>velocity boundary condition type for boundary surface N; may be:</p> <p>-6: expansion cell;</p> <p>-5: continuative mass flow;</p> <p>-4: uniform continuative velocity;</p> <p>-3: free slip (symmetry): $u_n(N) = 0, \tau_w = 0$;</p> <p>-2: continuative velocity;</p> <p>-1: continuative mass flux;</p> <p>1: specified constant velocity; $u_n(N) = u_{n0}(N)$;</p> <p>100+nf: specified transient velocity as function of time: $u_n(N) = u_{n0}(N)*f_{nf}(t)$;</p> <p>200+nf: specified transient velocity as function of temperature of radiation surface nf: $u_n(N) = u_{n0}(N)*f_{nf}(T_{r,nf})$.</p>	<p>$1 \leq N \leq \text{NSURF}$</p> <p>KTEMP(N)=400; May only be used on 10 surfaces.</p> <p>KPRES(N)=0</p> <p>KPRES(N)=0</p> <p>KPRES(N)=0 $1 \leq nf \leq 25$</p> <p>KPRES(N)=0 $1 \leq nf \leq 25$</p>
VELOC(N)	<p>u_{n0}: velocity [m/s] in the direction specified by the unit surface-normal vector for boundary surface N.</p>	<p>$1 \leq N \leq \text{NSURF}$</p> <p>KFLOW(N)>0 (VELOC(N)≥0)</p>
KPRES(N)	<p>pressure boundary condition type for boundary surface N; may be:</p> <p>0: none;</p> <p>1: specified constant pressure: $P_b(N) = P_{b0}(N)$;</p> <p>100+nf: specified transient pressure as function of time: $P_b(N) = P_{b0}(N)*f_{nf}(t)$.</p>	<p>$1 \leq N \leq \text{NSURF}$</p> <p>KFLOW(N)=-3 or >0</p> <p>KFLOW(N)<0 but ≠-3</p> <p>KFLOW(N)<0 but ≠-3 $1 \leq nf \leq 25$</p>
PRES(N)	<p>P_{b0}: pressure [Pa] for boundary surface N.</p>	<p>$1 \leq N \leq \text{NSURF}$</p> <p>KPRES(N)>0</p>

Table 4.5 Boundary Condition Variables in NAMELIST DATA (Cont'd)

Variable	Meaning/Value	Restrictions
KTEMP(N)	<p>energy boundary condition type for boundary surface N; may be:</p> <p>1: specified constant temperature: $T_b(N) = T_{b0}(N);$</p> <p>100 + nf: specified transient temperature as function of time: $T_b(N) = T_{b0}(N) * f_{nf}(t);$</p> <p>200: specified constant heat flux: $q_b(N) = q_{b0}(N);$</p> <p>300 + nf: specified transient heat flux as function of time: $q_b(N) = q_{b0}(N) * f_{nf}(t);$</p> <p>400: adiabatic: $q_b(N) = 0;$</p> <p>500 + nf: duct wall: additional input is required;</p> <p>600 + m: recirculation: temperature is set to that of surface m: $T_b(N) = \text{avg } [T_b(m)].$</p>	<p>$1 \leq N \leq \text{NSURF}$</p> <p>$1 \leq nf \leq 25$</p> <p>$1 \leq nf \leq 25$</p> <p>$0 \leq nf \leq 25$</p> <p>$1 \leq m \leq \text{NSURF}$ KTEMP(m) = 400</p>
TEMP(N)	<p>energy boundary condition reference value for boundary surface N; either:</p> <p>T_{b0}: temperature [$^{\circ}\text{C}$]; or</p> <p>q_{b0}: heat flux [W/m^2] from the boundary to the fluid.</p>	<p>$1 \leq N \leq \text{NSURF}$</p> <p>$\text{KTEMP}(N) < 200$</p> <p>$200 \leq \text{KTEMP}(N) < 400$</p>

Table 4.6 Additional¹ Energy Boundary Condition Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
MATWAL(N)	wall material index number for boundary surface N.	If KTEMP(N)=500+nf: $1 \leq \text{MATWAL}(N) \leq \text{NMATER}$ If KTEMP(N)<400: $0 \leq \text{MATWAL}(N) \leq \text{NMATER}$
WALLDX(N)	L_w : wall thickness [m] for boundary surface N.	If MATWAL(N)>0: $\text{WALLDX}(N) > 0$
IHTWAL(N)	m: correlation index number used to calculate fluid-wall heat transfer coefficient, U_f , for boundary surface N; may be: m=0: conduction in fluid; $1 \leq m \leq \text{NHEATC}$: standard form m; $51 \leq m \leq 99$: user form m. (Caution: IHTWAL(99) is used to select implicitness of coupling in thermal structure option; cf. Sec. 5.4.)	If KTEMP(N)=500+nf: $\text{IHTWAL}(N) > 0$
HYDWAL(N)	D_b : characteristic length [m] used in fluid-wall heat transfer coefficient for boundary surface N.	If IHTWAL(N)>0: $\text{HYDWAL}(N) > 0$
TSINK(N)	T_s : sink temperature [°C] on nonfluid side of boundary surface N.	
HSINK(N)	U_s : wall-sink heat transfer coefficient [$\text{W}/\text{m}^2 \cdot ^\circ\text{C}$] for boundary surface N.	
WALLQS(N)	q_{w0}'' : reference volumetric heat source [W/m^3] in wall for boundary surface N. The volumetric heat source in each wall segment is given by: $q_w'' = \text{QIJ}(i,j) * \text{QK}(k) * q_{w0}'' * f_{nf}(t).$	
QK(K)	relative power [-] in axial node K. (Caution: QK is used as shape multiplier for duct walls, internal fluid, thermal structure, and hexagonal fuel bundle heat sources.)	$1 \leq K \leq \text{KMAX}$ $\text{WALLQS}(N) \neq 0$

Table 4.6 Additional¹ Energy Boundary Condition Variables in NAMELIST DATA (Cont')

Variable	Meaning/Value	Restrictions
IFROD	indicator for reading NAMELIST INPUTQ; may be: 0: not read; 1: read. (Caution: IFROD > 0 has other meanings for hexagonal fuel bundle option, as discussed in Sec. 10.)	ISTATE=0 or 2
NOFQT	transient function number for internal heat generation in fluid.	$0 \leq \text{NOFQT} \leq 25$

Notes:

1. Input must be provided for the first 7 arrays shown in this table for boundary surfaces ($1 \leq N \leq \text{NSURF}$) having $\text{KTEMP}(N) = 500 + \text{nf}$; values input for these arrays are ignored for surfaces having $\text{KTEMP}(N) = 400$ or $600 + \text{m}$. Input may optionally be provided for the first 4 arrays shown in this table for surfaces having $1 \leq \text{KTEMP}(N) < 400$.

Table 4.7 Boundary Condition Variables in NAMELIST INPUTQ¹

Variable	Meaning/Value	Restrictions
QSCOOL	average volumetric heat source [W/m^3] in fluid (cf. Eq. 4.20b) when thermal structures are being used.	ISTRUC > 0
QIN(IND)	relative power [-] for cell position (I,J) where $\text{IND} = I + \text{IMAX} * (J - 1)$.	$1 \leq \text{IND} \leq 100$ $1 \leq \text{IMAX} \leq 10$ $1 \leq \text{JMAX} \leq 10$ See note 2.
QK(K)	relative power [-] in axial node K; the values of QK(K) must be positive for $\text{KLHS} \leq K \leq \text{KHHS}$.	$1 \leq K \leq \text{KMAX}$ See note 2.
KLHS	lowest heated K plane	$1 \leq \text{KLHS} \leq \text{KHHS}$
KHHS	highest heated K plane	$\text{KLHS} \leq \text{KHHS} \leq \text{KMAX}$
NOFQT	transient function number for internal heat generation in fluid.	$0 \leq \text{NOFQT} \leq 25$

Notes:

1. Only read if IFROD > 0 in NAMELIST DATA.
2. Arrays QIN and QK are used as shape multipliers for duct wall, internal fluid, thermal structure, and hexagonal fuel bundle heat sources. The values input will be normalized by the program.

the value entered in array TEMP is the temperature, which is then used to fill the proper locations in boundary surface array TLB; for KTEMP=200 and 300+nf, the value entered in array TEMP is the heat flux, which is then used to fill the proper locations in boundary surface array QBN. Zeros may be entered in arrays VELOC, PRES, and TEMP as "position fillers" for boundary surface types for which no entry is required (i.e. where "n.a." is shown in Tables 4.1 through 4.3); for example, if the velocity boundary conditions are specified-constant-value on surfaces 1 (at 10 m/s) and 4 (at 5 m/s) and continuative on surfaces 2 and 3, the user may use either

VELOC(1)=10., VELOC(4)=5.,

or

VELOC=10.,0.,0.,5.,

to enter values into the VELOC array. If the velocity or energy boundary condition is a specified-constant-value type, then 'VELB', 'TLB', and 'QBN' records in the Boundary Value Initialization input block may be used to enter spatially varying values of velocity, temperature, and heat flux to override the uniform distributions set by VELOC and TEMP; the format of these records is given in Sec. 4.2.2. (A similar option for setting pressures is not provided, since a specified-value pressure boundary condition must provide a value which is constant in space over the boundary surface area.)

The volumetric internal heat generation rate, if used, is also input at the beginning of a steady-state calculation. If there are no thermal structures (ISTRUC=0), then the reference values QSOUR(i,j,k) must be entered for each spatial location (i,j,k) using 'QSOU' records in the Internal Cell Initialization input block (cf. Sec. 4.2.2). If thermal structures are being used anywhere in the system volume (ISTRUC>0), then the heat source must be described using input variables QSCOO, QIN, and QK. Values for variable QSCOO and arrays QIN and QK are entered using NAMELIST INPUTQ as shown in Table 4.7. The user must specify a value for variable IFROD in NAMELIST DATA (cf. Table 4.6) which is greater than zero in order for NAMELIST INPUTQ to be read. (Input variable IFROD is also used in connection with the hexagonal fuel bundle model described in Sec. 10.)

At the beginning of a transient (ISTATE=2), specified-value type boundary conditions may be changed from constant to transient by altering the entries in the KFLOW, KPRES, and KTEMP arrays for the appropriate boundary surfaces. The internal heat generation may be time dependent if thermal structures are not being used by entering NOFQT > 0 in either NAMELIST DATA or INPUTQ (cf. Tables 4.6 or 4.7). The user must also input a transient function for each value of "nf" used in the KFLOW, KPRES, and KTEMP arrays and corresponding to the value of NOFQT (if NOFQT > 0). The transient functions are input using arrays TVAL, FVAL, and NEND as described in Sec. 4.1.4.

Certain other boundary condition changes may be made at the beginning of a transient. Generally, these changes must be physically logical; for example, a specified-value velocity boundary may be changed to a continuative velocity boundary condition combined with a specified-value pressure boundary condition (and vice versa) and an energy boundary condition may be changed from specified temperature to specified heat flux (and vice versa). On the other hand, the value of a specified-value boundary condition may not be changed at the beginning of a transient by specifying a new value in array VELOC, PRES, or TEMP; this would be a nonphysical change; instead, (a) the steady-state calculation should be rerun from ISTATE=0 with the proper value or (b) a transient function used to make a transition from the old value to the new value over a short period of time during the transient calculation.

Certain energy boundary conditions have additional input variables as shown in Table 4.6. If KTEMP < 400, the user may input a wall heat transfer coefficient correlation index number in array IHTWAL (cf. text after Eq. 4.13); if IHTWAL > 0, the characteristic length must be input in the corresponding location in array HYDWAL. If KTEMP < 400, the user may input a wall material index number in array MATWAL (cf. text after Eq. 4.14); if MATWAL > 0, the wall thickness must be input in the corresponding location in array WALLDX. If KTEMP = 500 + nf, the user must input nonzero values in arrays IHTWAL, MATWAL and WALLDX plus the temperature of the surroundings in array TSINK, the wall-to-surroundings heat transfer coefficient in array HSINK, and the volumetric heat source in the wall in array WALLQS. If WALLQS ≠ 0, the user may enter values in arrays QK and QIN in NAMELIST INPUTQ (cf. Table 4.7) to impose a spatial distribution on the heat source. (If NAMELIST

INPUTQ is being read ($IFROD > 0$), array QK must be entered in NAMELIST INPUTQ rather than in NAMELIST DATA). The user is cautioned that arrays QIN and QK are the spatial distribution factors for all types of heat sources in a calculation.

The input describing boundary conditions should be removed from the input stream for runs which continue a steady-state ($ISTATE=1$) or a transient ($ISTATE=3$) calculation.

4.2 Initial Conditions

Initial values must be assigned to all fluid variables before a calculation may be started; these include the state variables (enthalpy, pressure, temperature, and density) and the three components of velocity. These initial values are merely guesses to start a steady-state calculation ($ISTATE=0$); a good set of initial values will shorten the time required by the code to find steady state. On the other hand, these values should be true initial conditions at the beginning of a transient calculation; these values should generally be the result of a steady-state COMMIX calculation. The way in which these values are input, as well as the values themselves, depends on the type of calculation being performed.

4.2.1 Initialization Options

A total calculational history for a simulation with COMMIX usually requires a sequence of computer runs. Calculational information is passed from one computer run to another computer run via the "restart file". A user-input variable, IFRES in NAMELIST GEOM, determines whether restart files are to be used in a given computer run. If $IFRES=1, 3$, or 4 , a restart file is written at the end of a computer run. This file contains sufficient information to allow restoration of all calculational variables to their prior values and is input to the next run in the sequence, which will have $IFRES=2$ or 3 . The next run may continue a calculation in the same mode (i.e. steady state ($ISTATE=1$) or transient ($ISTATE=3$)) or begin a transient from a converged steady-state calculation ($ISTATE=2$). The restart file, thus, provides the initial conditions for all but the first run in a sequence.

For the first run in a sequence, the code performs a base level initialization as follows. Values of velocity, pressure, and temperature or heat flux on boundaries are initialized using any specified-value type boundary conditions as described in Sec. 4.1; the initial values of these quantities are not needed on other boundary surfaces (and they will be zero). The remainder of this section applies to initialization of nonboundary values. The user specifies reference values for temperature (T_{ref}) and for the pressure (P_{ref}) at one location ($x_{ref}, y_{ref}, z_{ref}$) for each fluid type m using input array locations TEMP0(m), PRES0(m), XPRES0(m), YPRES0(m), and ZPRES0(m), respectively, in NAMELIST DATA. (Note that arrays TEMP0 and PRES0, which are used to input reference conditions for each fluid are different from arrays TEMP and PRES, which are used to enter values for boundary conditions for each surface.) The reference temperature becomes the initial uniform value for all cells containing that fluid type. The reference temperature and pressure are used to establish a reference fluid density (ρ_{ref}) from the equation of state for each fluid. Using these values and the three components of the gravitational acceleration vector (g_i), the code will establish a reference state pressure for all fluid cells of each fluid type, using

$$P_0 = P_{ref} + \rho_{ref} [g_x (x - x_{ref}) + g_y (y - y_{ref}) + g_z (z - z_{ref})], \quad (4.24)$$

where x , y , and z are the Cartesian coordinates of the centers of the main (as opposed to momentum) control volumes; this pressure is stored in array PSTAT0. (The components of the gravitational acceleration vector (cf. Sec. 3.2) and the coordinates of the reference point are always specified on a Cartesian-coordinate basis, even when the cylindrical geometry option is selected.) The initial values for fluid velocities on all internal cell faces are zero. (The velocities on cell faces which lie on boundary surfaces having specified-value-type velocity boundary conditions are initialized using the values in the VELOC array as described in Sec. 4.1.)

After this base level initialization has been performed, the user has three ways to provide alternate values: restart guess, pointwise, and one-dimensional calculation. The first two of these options are discussed in the next two paragraphs. The one-dimensional calculation is part of the hexagonal fuel assembly option and is discussed in Sec. 10.

If the fluid conditions are expected to be similar to those in a prior run, the restart file written by the prior run may be input to the new run; use of this option is selected by setting $IFRES=4$ in new run's input. In this case, the temperature, pressure, and velocity distributions in the fluids will be initialized to the values read from this file. The fluid cell orderings for the old and new runs must be identical for this option to work properly; in particular, the two runs must have (1) the same number of fluid cells, boundary surfaces, and boundary surface elements and (2) identical records in the Boundary Surface Identification input block; use of this option is specifically disallowed by the code if the values of input variables $IGEOM$, $NM1$, $NL1$, $IMAX$, $JMAX$, and $KMAX$ are not identical for the old and new runs. This method of using a restart file as a guess differs from the normal use of a restart file to continue a calculational history: in the guess mode, only the temperature, pressure, and the three velocity components are initialized based on information read from the restart file; in the continuation mode, all variables are restored to their prior values.

The user also has the option to specify values on a pointwise basis using the Internal Cell Initialization input block. Values may be specified for fluid temperature, pressure, and the three components of velocity for as many grid locations as desired. These values are read after the optional reading of a guess from the restart file and override values read from the restart file.

After the base level initialization has been performed and optionally modified by the restart-guess and pointwise-input options, initial values for enthalpy and density for each fluid cell will be calculated from the appropriate equation of state using the initial values of temperature and pressure.

4.2.2 User Input

The user specifies initial conditions using variables in NAMELISTs $GEOM$ and $DATA$ and the Boundary Value Initialization and Internal Cell Initialization input blocks at the beginning of a steady-state calculation ($ISTATE=0$). The various input variables are shown in Tables 4.8 through 4.11.

Table 4.8 Initial Condition Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions ¹
TEMPO(NI)	T_{ref} : initial temperature [°C] for cells containing fluid number NI.	$1 \leq NI \leq IFHTX$ $TEMPO(NI) \neq 0$
PRESO(NI)	P_{ref} : reference pressure [Pa] at location $(x_{ref}, y_{ref}, z_{ref})$ for cells containing fluid number NI.	$1 \leq NI \leq IFHTX$
XPRESO(NI)	x_{ref} : x coordinate [m] of pressure reference point for cells containing fluid number NI.	$1 \leq NI \leq IFHTX$
YPRESO(NI)	y_{ref} : y coordinate [m] of pressure reference point for cells containing fluid number NI.	$1 \leq NI \leq IFHTX$
ZPRESO(NI)	z_{ref} : z coordinate [m] of pressure reference point for cells containing fluid number NI.	$1 \leq NI \leq IFHTX$

Notes:

1. These variables should only be specified at ISTATE=0.

The base level initialization at ISTATE=0 is controlled by the input arrays TEMPO, PRESO, XPRESO, YPRESO, and ZPRESO in NAMELIST DATA. As noted in Table 4.8, values must be supplied in these arrays for each fluid being used in the system volume.

The base level initialization will be overridden by values of velocity, pressure, and temperature read from the restart file under the restart-guess option if IFRES=4 in NAMELIST GEOM as shown in Table 4.9.

The values initialized by the base level and restart-guess options may be overridden on a pointwise basis using the Internal Cell Initialization input block. The fixed-format record structure of this input block is shown in Table 4.10. The first field indicates which array is to be initialized, the second field indicates the value to be assigned, and the third through eighth fields indicate a range of spatial locations. The indexing specified by IB, IE, JB, JE, KB, and KE depends on the quantity being initialized: the indices refer to the faces of computational cells for the three velocity components but refer to the centers of computational cells for the

Table 4.9 Initial Condition Variables in NAMELIST GEOM

Variable	Meaning/Value	Restrictions
IFRES	<p>restart file usage indicator; may be:</p> <p>0: restart file is not used;</p> <p>1: restart file is written at end of this computer run;</p> <p>2: restart file is read at the beginning of this computer run;</p> <p>3: restart file is read at the beginning and written at the end of this computer run;</p> <p>4: part of a restart file is read at the beginning of this computer run to initialize a guess for velocity, pressure, and temperature; a restart file will be written at the end of computer run.</p>	<p>ISTATE=0</p> <p>ISTATE ≥ 0</p> <p>ISTATE > 0</p> <p>ISTATE > 0</p> <p>ISTATE=0 Geometry of old and new problems must be identical; see text.</p>

other quantities in Table 4.10. Some sort of guess at the dominant structure of the velocity field should be entered using the 'UL ', 'VL ', and 'WL ' records in this input block when all boundary surfaces have something other than specified-(nonzero) value velocity boundary conditions. Note that any pressures entered using this block are on a relative basis, whereas the pressures entered using the PRES and PRES0 arrays are on an absolute basis. The total pressure in any computational cell in the code is the sum of two components and is given by

$$(P_{tot})_{i,j,k} = (P_0)_{i,j,k} + (P_{rel})_{i,j,k} \quad (4.25a)$$

$$(P_{tot})_{i,j,k} = [\text{array element PSTAT0}(i,j,k)] + [\text{array element } P(i,j,k)]. \quad (4.25b)$$

The quantity P_0 is the reference-state pressure; it is defined by Equation 4.24 in terms of the user input shown in Table 4.8 and is stored in array PSTAT0. The other component, P_{rel} , is the total pressure relative to P_0 ; it is stored in array P; the default initial value is zero and nondefault initial values may be entered by the user via the 'P ' records in the Internal Cell Initialization

Table 4.10 Initial Condition Variables in Internal Cell Initialization Input Block

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	<p>card identifier; may be:</p> <p>'P ': "VALUE" is pressure [Pa] as a value relative to P_0;</p> <p>'QSOU': "VALUE" is volumetric heat source, QSOUR, [W/m³]; (Use QSCOO, QIN, and QK if ISTRUC>0.)</p> <p>'TL ': "VALUE" is temperature, T, [°C];</p> <p>'UL ': "VALUE" is x- or r-component of velocity, u, [m/s];</p> <p>'VL ': "VALUE" is y- or θ-component of velocity, v, [m/s];</p> <p>'WL ': "VALUE" is z-component of velocity, w, [m/s];</p> <p>'END ': terminates this input block; record positions 5-38 should be blank.</p>	ISTRUC=0 (If IFROD>0, these values override those set by QFLUX, QK, and QIN in NAMELIST INPUTQ.)
5-14	F10.4	VALUE	See NAME.	
15-18	I4	IB	beginning I-range index.	$1 \leq IB \leq IE$
19-22	I4	IE	ending I-range index.	$IB \leq IE \leq MAX$
23-26	I4	JB	beginning J-range index.	$1 \leq JB \leq JE$
27-30	I4	JE	ending J-range index.	$JB \leq JE \leq JMAX$
31-34	I4	KB	beginning K-range index.	$1 \leq KB \leq KE$
35-38	I4	KE	ending K-range index.	$KB \leq KE \leq KMAX$
39-80	n.a.	n.a.	user comment (ignored).	Optional.

Table 4.10 Initial Condition Variables in Internal Cell Initialization Input Block (Cont'd)

Notes:

1. These records result in assigning the value "VALUE" to the variable named "NAME" for the spatial positions (I,J,K) in the three-dimensional region delimited by $IB \leq I \leq IE$, $JB \leq J \leq JE$, and $KB \leq K \leq KE$ within the system volume.
2. Use as many 'P ', 'QSOU', 'TL ', 'UL ', 'VL ', and 'WL ' records as needed to provide pointwise initializations of the indicated variables which override the automatic initializations performed by the code; the order of the records is irrelevant, except if a cell or cell face is assigned a value in more than one record, the final such specification will be used.
3. For 'P ', 'QSOU' and 'TL ', the indices (I,J,K) refer to cell-center grid points; the cells must actually exist (i.e. contain fluid).
4. For 'UL ', the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I+1,J,K).
5. For 'VL ', the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I,J+1,K).
6. For 'WL ', the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I,J,K+1).
7. For 'UL ', 'VL ', and 'WL ', the cell faces referenced must exist and must not lie on boundary surfaces.
8. These 'P ', 'QSOU', 'TL ', 'UL ', 'VL ', and 'WL ' records may be interspersed among the other records (i.e. 'AL ', 'ALX ', 'ALY ', 'ALZ ', 'TURC' and 'TURV') in the Internal Cell Initialization Input Block (cf. Tables 3.10 and 8.3). An 'END ' record terminates the input block, occurs only once, and must be last.

input block. The corresponding elements in these two arrays are summed by the code to get the total pressure when needed. (There is a substantial numerical advantage to storing pressure as the sum of these two components in the finite-difference formulation used in COMMIX.) The 'QSOU' records in this input block are used to define the source of internal heat generation in the fluid (cf. Eq. 4.20a) when thermal structures are not being used in the calculation.

The Boundary Value Initialization input block is used to establish a spatial distribution of values when specified-constant-value velocity and energy boundary conditions are being used. The fixed-format record structure of this input block is shown in Table 4.11. The first field indicates which array is to be initialized, the second field indicates the value to be assigned, the third through eighth fields indicate a range of spatial locations, and the ninth field indicates the boundary surface number. The values entered via 'VELB', 'TLB ', or 'QBN ' records are used directly to fill locations in arrays VELBN, TLB, or QBN, respectively, and override the uniform values established via the input arrays VELOC and TEMP when the surface has corresponding entries of KFLOW=1, KTEMP=1, or KTEMP=200. Values for these variables should not be input using these records for boundary surfaces having other types of boundary conditions. The velocity entered using a 'VELB' record (or using the VELOC array) is defined relative to

Table 4.11 Boundary Value Initialization Input Block Record Structure

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier; may be: 'AREA': "VALUE" is flow area [m ²]; this is normally input in Boundary Surface Identification input block; 'QBN ': "VALUE" is heat flux, q_b , [W/m ²] from the boundary to the fluid; 'TLB ': "VALUE" is temperature, T_b , [°C]; 'VELB': "VALUE" is velocity, u_b , [m/s] in direction of unit surface-normal vector; 'END ': terminates this input block; record positions 5-42 should be blank.	 KTEMP(N)=200 KTEMP(N)=300+nf KTEMP(N) < 200 KTEMP(N) > 400 KFLOW(N)=1
5-14	F10.4	VALUE	see NAME.	
15-18	I4	IB	beginning I-range index.	$1 \leq IB \leq IE$
19-22	I4	IE	ending I-range index.	$IB \leq IE \leq IMAX$
23-26	I4	JB	beginning J-range index.	$1 \leq JB \leq JE$
27-30	I4	JE	ending J-range index.	$JB \leq JE \leq JMAX$
31-34	I4	KB	beginning K-range index.	$1 \leq KB \leq KE$
35-38	I4	KE	ending K-range index.	$KB \leq KE \leq KMAX$
39-42	I4	N	boundary surface number.	$1 \leq N \leq NSURF$
43-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. These records result in assigning the value "VALUE" to the variable named "NAME" for surface elements (I,J,K), on boundary surface N of the system volume. A boundary surface element is referenced by the indices (I,J,K), where $IB \leq I \leq IE$, $JB \leq J \leq JE$, and $KB \leq K \leq KE$, of the cell which is adjacent to the boundary surface element and toward which the unit surface-normal vector points.
2. Use as many 'AREA', 'QBN ', 'TLB ', and 'VELB' records as needed to provide pointwise initializations of the indicated variables which override the automatic initializations performed by the code; the order of the records is irrelevant, except (a) if a boundary surface element is assigned a value in more than one record, the final such specification will be used and (b) the 'END' record only occurs once and is last.

the direction of the unit surface-normal vector (which points from the boundary toward the fluid system): flow of fluid into the system across the boundary has a positive velocity. Similarly, the heat flux entered using a 'QBN' record (or using the TEMP array) is defined as positive for heat flow from the boundary surface to the fluid.

The input described in Tables 4.8, 4.10, and 4.11 should be removed from the card-image input file when continuing a steady-state calculation (ISTATE=1), beginning a transient calculation from a steady-state calculation (ISTATE=2), and continuing a transient calculation (ISTATE=3). All of the information supplied by these variables must be obtained from the restart file in order to maintain a numerically consistent calculation and should not be overridden. (Some of the input, if left in the input stream, is harmless -- it will either make no change or will be ignored; leaving some other input variables in the input stream may introduce an unwanted discontinuity.) There is one important exception: the 'END' record in both the Boundary Value Initialization and Internal Cell Initialization input blocks must always be present in the card-image input file even though no values are being specified for the various variables. (Of course, the user must specify a value of 2 or 3 for variable IFRES in NAMELIST GEOM for the input restart file to be read.)

5. THERMAL STRUCTURES

The thermal behaviors associated with solids embedded in the fluid in the system volume are governed by the thermal structure model. Examples of solids for which the thermal structure model might be used include the fuel pins in a nuclear reactor, the tube bundle in a heat exchanger, flow baffle plates, pipes, and the barrier separating two different fluids. The modeling provides for calculation of (a) conduction-based temperature distributions within these solids, which may contain heat sources, (b) exchange of energy with the fluid in the system volume via convection, and (c) exchange of energy among the various solids via radiation. The equations for this modeling are given in Sec. 6 of the Equations and Numerics volume (Ref. 1); the features of the model are summarized below in Sec. 5.1.

Use of the thermal structure model is optional, and, thus, the associated input is optional. The input for the thermal structure model is contained in five different sections of the text input file: the Thermal Structure Prototype block, the Thermal Structure Location block, and NAMELISTs GEOM, DATA, and STRUCT. The first two of these sections are described in Secs. 5.2 and 5.3; the remainder of the input is described in Sec. 5.4.

5.1 Review of Model Features

A thermal structure has one of three basic shapes: slab, cylinder, or sphere. The unit surface-normal vector of a slab-type thermal structure and the axis of a cylindrical-type thermal structure must be aligned with one of the three fluid coordinate axes (i.e. x (or r), y (or θ), or z). By way of illustration, Fig. 5.1 shows a cylindrical thermal structure whose axis is aligned with the z axis of the fluid grid.

Intersection of the fluid grid coordinates with thermal structures divides the structure into "thermal structure elements". (This is only strictly true for slabs and cylinders; a sphere is treated differently, as will be discussed in Sec. 5.3.) For the example shown in Fig. 5.1, the intersection of the fluid's axial coordinate grid divides the structure into thermal structure elements having a circular cross section and height Δz .

Fig. 5.1 Definition of a Cylindrical Thermal Structure Element Relative to the Flow Domain

Each thermal structure element has two "surface elements": outer and inner. It is only through these two surface elements that a thermal structure element exchanges heat with a fluid (by convection) or with another thermal structure surface element (by radiation). One of these two surface elements may be adiabatic; this is always the case for the inner surface of a solid cylinder or sphere. Various arrangements of thermal structure surface elements are shown in Fig. 5.2. Each surface element may interact with only one fluid cell; using the example in Fig. 5.3, what appears to be a single cylinder interacting with four fluid cells must actually be modeled as four quarter-cylinder thermal structure elements, each of which interacts with a single fluid cell. On the other hand, a single fluid cell may interact with surface elements of several thermal structures, as illustrated in Fig. 5.4.

Fig. 5.2 Outer and Inner Surfaces of Several Thermal Structure Elements in the Fluid Grid Structure

Fig. 5.3 Four Quarter-Cylindrical Thermal Structure Elements, Each Interacting with One Fluid Cell

The specification of which fluid cell interacts with each surface element of each thermal structure element determines the physical location of a thermal structure within the system volume. No further specificity is employed which would, for example, locate the three thermal

structures shown in Fig. 5.4 at the specific locations shown within the cell; the formulation in COMMIX only needs to know that the surfaces of these three thermal structures all interact with the same cell. If both surfaces of a thermal structure element interact with fluid, then these should be two different fluid cells. (In this regard, the geometry shown in Fig. 5.2b is only allowed if the inner surface is adiabatic.) The two fluid cells are normally adjacent to each other in the grid, but this is not a requirement.

The space between the two surface elements is composed of one or more "regions" of solid materials in the form of layers. This is illustrated for a cylindrical thermal structure element in Fig. 5.5; the outer and inner surface elements are denoted surfaces 1 and 2, respectively, in this figure. Each material region is composed of one or more equal-size "partitions". Material regions are separated from each other by a "gap" which is void of material.

An energy equation is formulated and solved in the material regions between the two surface elements; in the finite-difference formulation, a temperature is calculated at the center (not the surface) of each material partition. The calculation is in one dimension, from the outer to the inner surface element; this direction is radial (with respect to the solid, not the fluid) for cylindrical and spherical thermal structures and in the direction of the unit surface-normal vector for planar thermal structures. Temperature is uniform in the other two directions within a single thermal structure element. A material region may have internal heat generation, which may be a function of time. Within a material region, heat is transferred from partition to partition by conduction. Heat transfer from one material region to the next material region is by convection across the gap. At the outer and inner surface elements, heat may be transferred to or from (a) the fluid by convection and/or (b) a surface element of another thermal structure by radiation; these are the only two surfaces through which energy enters or leaves the thermal structure element.

Thermal structure surface elements may exchange energy by thermal radiation with other thermal structure surface elements. This is accomplished by defining a set of "radiation surfaces". Each radiation surface is composed of one or more thermal structure surface elements for which the thermal radiation heat transfer characteristics are similar. The temperature of a

Fig. 5.4 Multiple Thermal Structures Interacting with a Single Fluid Cell

Fig. 5.5 Material Regions and Gaps for a Typical Thermal Structure Element

radiation surface is considered to be spatially uniform; this temperature is taken to be the area-weighted average of the temperatures of the thermal structure surface elements of which it is composed. The net radiation heat flux is computed for each radiation surface using this average temperature; the resulting heat flux is then assigned uniformly to each thermal structure surface element which forms the radiation surface and, thus, becomes a boundary condition for the energy calculation being performed within a thermal structure element. If the use of a uniform heat flux for each surface element is felt to be an inadequate approximation, the user should divide the physical structure into smaller pieces for which the uniformity assumption is adequate and model each of these pieces as a different thermal structure. All radiation surfaces are assumed to be gray and opaque. Emissions from and reflections by radiation surfaces are diffuse and independent of wavelength. The intervening material between two radiation surfaces is perfectly transparent.

5.2 Thermal Structure Prototype Input Block

The concept of a "thermal structure prototype" is used to define a thermal structure in terms of the following physical characteristics: basic shape, orientation, multiplicity per fluid cell, material(s) of composition, physical size (of each material region), internal heat source, and surface boundary conditions. If these physical characteristics are all the same for several objects in the physical system, they may be described to the code using a single thermal structure prototype. Thus, a physical system having many submerged solids can typically be modeled using a number of thermal structure prototype definitions which is less than the actual number of physical objects (although this may still mean that many thermal structure prototypes will be needed). The definition of thermal structure prototypes is accomplished using the Thermal Structure Prototype input block, as described in this section. The placement of thermal structures within the fluid system is described to the code by specifying which fluid cells interact with the surfaces of each thermal structure prototype; this is accomplished using the Thermal Structure Location input block, as described in Sec. 5.3. A thermal structure prototype, in essence, defines a single thermal structure element; the Thermal Structure Location input block places this basic thermal structure element at specific locations in the fluid grid system, allowing for replication of the same type of object at multiple locations.

The structure of the Thermal Structure Prototype input block is shown in Table 5.1. The input for each prototype is a collection of three different NAMELISTs: T, F, and M, which may be remembered as specifying the type of, fluid interactions with, and material regions comprising, respectively, a thermal structure prototype. The input for the first prototype is followed by input for the second, third, etc. prototypes until all prototypes have been defined. The final entry in this input block is 'END ' in record positions 1-4 to signal the end of the block. The number of thermal structure prototypes being defined is determined by the code from the number of occurrences of NAMELIST T in this input block.

The information to be provided in NAMELISTs T, F, and M is described in the next three subsections. As noted in Table 5.1, the order in which these records appear is variable depending on the characteristics of each thermal structure prototype. Of particular note is the way in which default values are used. The NAMELISTs in this block follow the general rule: if a value is not explicitly specified by the user for a variable, then the variable retains its "current" value. For the first time a given NAMELIST appears in this input block, the "current" value is the default value as set by the code; for subsequent occurrences of the same NAMELIST, the "current" value is the value in effect after reading the prior NAMELIST and is, thus, the original default value only up to the point in this input block at which the user explicitly specifies a nondefault value in the NAMELIST. Stating this another way, the "current" values of variables are not reset to the original default values just prior to reading each new NAMELIST. For example, if all thermal structure prototypes are cylinders aligned with the z axis, then the user must specify IXYZ=3 in the first NAMELIST T but may omit specifying a value for this variable in subsequent NAMELIST T's; a safer practice is to always explicitly specify values for variables rather than allowing values to propagate in this manner. The single exception to the no-reset rule is for input variable KRAD, which identifies radiation surfaces; the "current" value of KRAD is reset to zero before reading each NAMELIST M.

Table 5.1 Thermal Structure Prototype Input Block Structure

Record Description	Restrictions
NAMelist T for first prototype.	
NAMelist F for outside surface of first prototype.	Outside interacts with fluid.
NAMelist M for material region next to outside surface of first prototype.	
..... A NAMelist M for each material region of first prototype, working from outside to inside.	If needed to define multiple regions.
NAMelist M for material region next to inside surface of first prototype.	If needed to define multiple regions.
NAMelist F for inside surface of first prototype.	Inside interacts with fluid.
NAMelist T for second prototype.	
NAMelist F for outside surface of second prototype.	Outside interacts with fluid.
NAMelist M for material region next to outside surface of second prototype.	
..... A NAMelist M for each material region of second prototype, working from outside to inside.	If needed to define multiple regions.
NAMelist M for material region next to inside surface of second prototype.	If needed to define multiple regions.
NAMelist F for inside surface of second prototype.	Inside interacts with fluid.
..... NAMelists T, F, M, and F for third, fourth, fifth, M-th prototype.	
NAMelist T for last prototype.	
NAMelist F for outside surface of last prototype.	Outside interacts with fluid.
NAMelist M for material region next to outside surface of last prototype.	
..... A NAMelist M for each material region of last prototype, working from outside to inside.	If needed to define multiple regions.
NAMelist M for material region next to inside surface of last prototype.	If needed to define multiple regions.
NAMelist F for inside surface of last prototype.	Inside interacts with fluid.
'END ' in record positions 1-4 to terminate Thermal Structure Prototype input block. (Record positions 5-80 may contain user comments, which will be ignored.)	Must be last.

Table 5.1 Thermal Structure Prototype Input Block Structure (Cont'd)

Notes:

1. This input block is present if, and only if, ISTRUC=2 and NEWTS=1, which is only permitted if ISTATE=0 or =2.
2. The beginning of a new NAMELIST must be indicated by having the string ' &T ', ' &F ', or ' &M ' in record positions 1-4.
3. Within a prototype, the order of records is significant and determines the composition order and surface interactions. Each prototype definition begins with a NAMELIST T and must additionally have at least one each of NAMELISTs F and M.
4. User comment records (denoted by blanks in record positions 1-4 and having the text of the comment in positions 5-80) may be present at the beginning of this block, between NAMELISTs in this block, and before the 'END' record.
5. If a value is not specified for a required input variable, the value specified for this variable in the most recent prior occurrence of the NAMELIST will be used.

5.2.1 NAMELIST T

The beginning of a new thermal structure prototype definition is denoted by the presence of NAMELIST T to identify the type of thermal structure. The variables to be input using NAMELIST T are shown in Table 5.2. Each thermal structure prototype must be assigned an index number using the input variable N. Although not a requirement, the user will typically find it most convenient to assign these index numbers in a contiguous ascending sequence beginning with 1. In any case, the value of N may not exceed the value of code-internal variable PNSTRU; typically, PNSTRU=99 but this may have been altered by the person who installed COMMIX at your site (cf. Appendix C.1.1). Also, a different value must be input for N in each NAMELIST T, as this index number is used to identify the thermal structure prototype.

Thermal structure prototypes have three basic shapes: slab, cylinder, and sphere. Additionally, the unit surface-normal vector of a slab and the axis of a cylinder must be aligned with one of the three coordinate axes (i.e. x (or r), y (or θ), or z). The combination of these shape and alignment properties is defined using input variable IXYZ in NAMELIST T. (The alignment for a sphere is only relevant if internal heat sources are to have a spatial distribution, which is discussed in Sec. 5.2.3 in connection with input variable Q in NAMELIST M.) For example, the cylindrical thermal structure aligned with the z-axis in Fig. 5.1 is described by

IXYZ=3. For cylindrical and spherical thermal structures, the outer radius of the solid must be input using variable OUTR; this radius is relative to the center of the thermal structure, which is not necessarily the same as the physical location of the surface with respect to the radial center of a cylindrical coordinate fluid grid system (i.e. when IGEOM=-1); the inner radius will be computed from OUTR and the information supplied in NAMELISTs M (cf. Sec. 5.2.3.).

The input variable RODFR allows multiple copies or fractional parts of a thermal structure prototype to be the basic form which interacts with fluid cells. The exact meaning of RODFR

Table 5.2 Thermal Structure Variables in NAMELIST T which Begins the Definition of a Prototype

Variable	Meaning/Value	Restrictions
N	thermal structure prototype number.	$1 \leq N \leq \text{PNSTRU}$
IXYZ	<p>geometry indicator, which is a three-digit integer of the form "TTD" (leading zeros may be omitted), where "TT" and "D" are defined below.</p> <p>TT: geometry type; may be:</p> <p>00: cylinder having axis aligned with direction D;</p> <p>01: slab having unit surface-normal vector aligned with direction D;</p> <p>10: sphere aligned with direction D.</p> <p>D: alignment direction (also sets the meaning of index m in heat source multiplier QK(m); may be:</p> <p>1: x or r direction (m is I);</p> <p>2: y or θ direction (m is J);</p> <p>3: z direction (m is K).</p>	IXYZ=1,2,3, 11,12,13, 101,102, or 103
OUTR	outer radius [m] of cylinder or sphere. (Ignored for TT=01.)	TT=00 or 10 OUTR > 0

Table 5.2 Thermal Structure Variables in NAMELIST T which Begins the Definition of a Prototype (Cont'd)

Variable	Meaning/Value	Restrictions
RODFR	<p>multiplicity of prototype; 'number' is not restricted to integers in what follows; meaning depends on IXYZ:</p> <p>>0: number of cylinders or spheres interacting with each associated fluid cell;</p> <p>>0: slab surface area [m²] interacting with each associated fluid cell;</p> <p><0: absolute value is the number of cylinders per unit area of each associated fluid cell; this unit area depends on D:</p> <p style="margin-left: 40px;"><u>D</u> Area _____</p> <p style="margin-left: 40px;">1 $\Delta y \Delta z$ or $r \Delta \theta \Delta z$</p> <p style="margin-left: 40px;">2 $\Delta x \Delta z$ or $\Delta r \Delta z$</p> <p style="margin-left: 40px;">3 $\Delta x \Delta y$ or $\Delta r r \Delta \theta$</p> <p>and is for the cell adjacent to the outside surface if both surfaces interact with fluid;</p> <p><0: absolute value is the slab area divided by the cell area of each associated fluid cell; the cell area depends on D as given for TT=00;</p> <p><0: absolute value is the number of spheres per unit volume (i.e., $\Delta x \Delta y \Delta z$ or $\Delta r r \Delta \theta \Delta z$) of each associated fluid cell.</p>	<p>RODFR $\neq$ 0</p> <p>TT=00 or 10</p> <p>TT=01</p> <p>TT=00</p> <p>TT=01</p> <p>TT=10</p>
NT	nf: transient function number which multiplies heat source.	<p>Q=0: NT=0</p> <p>Q$\neq$0: $0 \leq NT \leq 25$</p>

depends on the shape of a thermal structure and the sign of RODFR. For cylindrical and spherical shapes, a positive value of RODFR specifies the "number" of copies of the thermal structure which will interact with each fluid cell; "number" is not restricted to being a whole number, and if it is between 0 and 1 then it specifies a fractional part of the basic shape. This concept is illustrated in Fig 5.6 for a cylindrical thermal structure; part (a) shows ten identical cylinders interacting with one fluid cell and is described using RODFR=10.0; part (b) shows one-quarter of a cylinder interacting with one fluid cell and is described using RODFR=0.25.

For cylinders and spheres, a negative value of RODFR has a meaning similar to a positive value of RODFR, except that the "number" of copies is on a per fluid cell area basis for cylinders and on a per fluid cell volume basis for spheres. This concept is illustrated in Fig. 5.7 for a cylindrical thermal structure prototype replicated uniformly in a Cartesian coordinate fluid grid, for which the value of RODFR is $-2/(a*b)$. Specifying $RODFR < 0$ is particularly useful for placing a cylindrical shell at a constant radial position in the fluid when using a cylindrical coordinate fluid grid having a nonuniform theta grid spacing; for the example shown in Fig. 5.8, the cylindrical thermal structure has its center at the radial center of the fluid grid system and the value of RODFR is $-1/(2*\pi*((r_2 + r_3)/2)*(r_3 - r_2))$ (if the outside surface interacts with fluid cells); it is by specifying RODFR in this manner that the proper size section of the cylinder is assigned to the fluid cell at each theta location. For a slab, a positive value of RODFR is the cross-sectional area, in square meters, per fluid cell; a negative value of RODFR expresses this area as a fraction or multiple of a cell face area. By way of example, a flat impervious plate which crosses through one or more cells of a fluid grid system which uses variable-size cell spacings is most easily described using $RODFR = -1.0$.

There are several important things to note when choosing a negative value for RODFR (cf. Table 5.2). The cell area upon which RODFR is based for cylindrical and planar thermal structures is a function of the structure's alignment direction. For all three thermal structure shapes, the cell area and cell volume upon which RODFR is based does not include a directional surface porosity or a volume porosity in the definition. If fluid interacts with both surfaces of a thermal structure, then the "per cell" used in the definitions for $RODFR < 0$ is for the fluid cell which interacts with the outside (not inside) surface of each thermal structure element; this is an important distinction when a nonuniform or cylindrical-coordinate fluid grid is being used.

The final variable allowed in NAMELIST T is NT. The value input for this variable is the index number of the transient function which will be used to multiply all internal heat sources in this thermal structure prototype. Further discussion of heat sources is provided in connection with input variable Q in NAMELIST M (cf. Sec.5.2.3) and NTOTS in NAMELIST DATA (cf. Sec. 5.4).

Fig. 5.6 Use of Nonunity RODFR Values in Assigning a Thermal Structure to a Cell:
(a) Multiple (RODFR=10.0) and (b) Fractional (RODFR=0.25)

Fig. 5.7 Use of a Negative RODFR Value ($= -2/(a*b)$) in Assigning a Uniform Distribution of a Thermal Structure in a Nonuniform Grid

Fig. 5.8 Use of a Negative RODFR Value ($= -1/(\pi*(r_3^2-r_2^2))$) in Assigning a Cylindrical Shell Thermal Structure in a Nonuniform Azimuthal Grid

5.2.2 NAMELIST F

The second type of input for a thermal structure prototype is given in NAMELIST F, which describes the interaction of fluid with the surface of a thermal structure. The input variables in NAMELIST F are shown in Table 5.3.

Table 5.3 Thermal Structure Variables in NAMELIST F which Defines Heat Transfer to Fluid from the Outside and/or Inside Surface of a Prototype

Variable	Meaning/Value	Restrictions
IHT	m: corelation index number used to calculate surface-fluid heat transfer coefficient, U_{cool} .	$1 \leq IHT \leq NHEATC$ $51 \leq IHT \leq 99$ See Sec. 9.3.
HYD	D_b : characteristic length [m] used in surface-fluid heat transfer coefficient.	$HYD > 0$

Heat transfer from a thermal structure to a fluid is by convection from the thermal structure surface element to the adjacent fluid cell using a heat transfer coefficient U_{cool} . (Since the finite-difference formulation in the code defines thermal structure temperatures at the centers of material partitions and fluid temperatures at the centers of fluid cells, this heat transfer near the surface is composed of conduction through a half thickness of the material partition at the surface followed by convection (using U_{cool}) from the surface to the center of the adjacent fluid cell.) The user supplies convection heat transfer coefficients as a set of one or more correlations involving local fluid conditions, as described in Sec. 9.3. The input in NAMELIST F selects which one of these correlations is to be used for this surface with the value of input variable IHT and supplies the characteristic length (D_b) as the value of input variable HYD.

A NAMELIST F immediately follows the NAMELIST T if fluid interacts with the outer surface of a thermal structure prototype; a NAMELIST F follows the final (or only) NAMELIST M if fluid interacts with the inner surface of a thermal structure prototype. Fluid must interact

with at least one of the two surface elements of a thermal structure element; therefore, at least one NAMELIST F must be present for each thermal structure prototype being defined. Fluid may interact with both sides of a thermal structure; therefore, there are, at most, two NAMELIST F's for each thermal structure prototype definition.

Several examples of outer and inner surface arrangements are shown in Fig. 5.2. Part (a) is a slab having a plane of symmetry in the middle; the basic thermal structure element is half of the slab thickness; this thermal structure element is duplicated in the cell using RODFR (e.g. RODFR=-2.0); the outside surface of the thermal structure element interacts with fluid and the inside surface, which is the plane of symmetry, does not interact with fluid. Part (b) is a (hollow) cylinder; the outside surface interacts with fluid; since the inside surface of this cylinder is in the same fluid cell, it must not interact with fluid (i.e. it is adiabatic). Part (c) is a slab where both the inside and outside surfaces interact with (different) fluid cells. Part (d) is a slab where only the outside surface interacts with fluid; the inside surface is adiabatic and, thus, does not interact with a fluid cell.

5.2.3 NAMELIST M

The third type of information defining a thermal structure prototype is given in NAMELIST M, which describes the material regions which compose the thermal structure. The input variables in NAMELIST M are shown in Table 5.4.

Each thermal structure prototype is composed of one or more material regions. Each NAMELIST M defines one of these material regions. The first NAMELIST M for a thermal structure prototype is for the material region which is adjacent to the outer surface. Subsequent occurrences of NAMELIST M for a thermal structure prototype define material regions which are progressively inward from the first (outer) material region specified.

Input variable MI defines the type of material. This index selects which set of user-defined solid material properties will be used to evaluate the density, thermal conductivity, and specific

Table 5.4 Thermal Structure Variables in NAMELIST M which Defines a Material Region within a Prototype

Variable	Meaning/Value	Restrictions
MI	material index number.	$1 \leq MI \leq NMATER$
NP	number of partitions.	See note 1.
DR	Δr : partition thickness [m].	See note 2.
Q	q_{00} : reference heat source per unit volume of partition [W/m^3]; the volumetric heat source is given by $q_v = q_{00} * QK(m) * f_m(t)$.	
KRAD	if nonzero, denotes that a 'surface' of this prototype is (part of) radiation surface number KRAD; 'KRAD' is also the value referred to by 'nf' in the $KFLOW = 200 + nf$ boundary condition.	$IFRAD > 0$ $0 \leq KRAD \leq NRAD$ See note 3.
HGAP	U_{gap} : gap heat transfer coefficient [$W/m^2 \cdot ^\circ C$].	$HGAP \neq 0$ See note 4.
SGAP	gap thickness [m].	$TT \neq 01$, $SGAP \geq 0$ See note 4.

Notes:

1. Sum of NP over all NAMELIST M's defining a prototype must not exceed 49.
2. Sum of $(NP * DR + SGAP)$ over all NAMELIST M's defining a cylindrical or spherical prototype must not exceed OUTF.
3. For each prototype, 'surface' means the outside surface if $KRAD > 0$ in the first (or only) NAMELIST M and the inside surface if $KRAD > 0$ in the final NAMELIST M; the value of KRAD is ignored in all other NAMELIST M's; there must be at least two NAMELIST M's for the inner surface to be a radiation surface.
4. For each prototype, the values of HGAP and SGAP are for the gap between this material region and the next (inward) material region; do not specify values in the final (or only) NAMELIST M.

heat capacity when solving the energy equation in this material region. These properties may be functions of temperature, and the input defining these sets of solid material properties is described in Sec. 9.2.

The physical size of a material region is input using variables NP and DR. The material

region will be divided into NP partitions, each of which has a thickness (Δr) of DR; the total thickness of the material region is, thus, NP*DR. The sum of NP over all NAMELIST M's used to define a thermal structure prototype must not exceed 49. For cylindrical and spherical thermal structure prototypes, the sum of [NP*DR + SGAP(see below)] for all material regions must be less than or equal to the value specified for OUTR in NAMELIST T; the solid is hollow if the sum is less than OUTR. A temperature is computed at the center of each one of these partitions in the finite-difference formulation used in the code.

A material region may contain internal heat generation; an example of this is a fuel rod in a nuclear reactor where the heat generation is the result of the nuclear fission process. The heat source, on a per volume of solid material in the thermal structure basis, is assumed to be in the form

$$q_i = q_0 * QK(m) * f_{nf}(t), \quad (5.1)$$

where q_0 is a reference value which is input using variable Q in NAMELIST M. The term QK(m) is a spatial shaping factor; the meaning of index m is determined by the rightmost digit of input variable IXYZ in NAMELIST T (i.e. "D" as shown in Table 5.2), and is either the I, J, or K spatial location index in the fluid grid coordinate system corresponding to the location of the thermal structure element. The factor QK(m) allows, for example, the internal heat generation rate in a fuel rod aligned in the z direction (IXYZ=3) to be a function of the axial position: the heat source would be multiplied by QK(1) for the section of the rod at axial location K=1, by QK(2) at K=2, ..., and by QK(KMAX) at K=KMAX. (For the heat generation rate to also be a function of radial position within the rod, multiple material regions with different Q values would be used to define the thermal structure prototype.) The array QK is input in NAMELIST DATA; this array is used for all orientations of all thermal structures and is also used to provide spatial shape information for other types of heat sources; in practice, a spatial variation can only be provided in one direction in any given calculation using array QK. The heat source in Eq. 5.1 is allowed to be time dependent, being multiplied by transient function number nf; input variable NT in NAMELIST T sets the value of nf and, thus, selects which one of the transient functions will be used. (The transient function assignments made using the values of NT in NAMELIST M are overridden by values specified using array NTOTS

in NAMELIST DATA, as discussed in Sec. 5.4.) The way in which transient functions are defined by the user to the code is described in Sec. 4.1.4. If the heat source is not a function of time, enter $NT=0$, and the code will replace $f_w(t)$ with 1.0.

Input variable KRAD is used to denote that one of the surfaces of this thermal structure prototype exchanges energy via thermal radiation with the surface of another thermal structure prototype(s). If $KRAD > 0$ in the first NAMELIST M of a prototype, then the outer surface elements of this prototype become part of radiation surface number KRAD. If $KRAD > 0$ in the last NAMELIST M of a prototype, then the inner surface elements of this prototype become part of radiation surface number KRAD; there must be at least two NAMELIST M's for the inner surface of a thermal structure to be a radiation surface. If the surface of a thermal structure is not to be part of a radiation surface, then enter $KRAD=0$ in the corresponding NAMELIST M. A nonzero value input for KRAD is ignored in all but the first and last NAMELIST M within a given thermal structure prototype definition. The value of KRAD is also the radiation surface number referred to by "nf" in the $KFLOW=200+nf$ velocity boundary condition (cf. Sec. 4.1.1). The same value may be specified for KRAD in more than one thermal structure prototype definition, if that makes sense in the physical system. The values entered for KRAD must form a contiguous sequence of integers beginning with 1; however, the values do not need to form a strictly increasing sequence as you move from the beginning to the end of the Thermal Structure Prototype input block; for example, if there are 5 radiation surfaces, they must be defined using KRAD values of 1 through 5, but radiation surface number 1 may be defined in a NAMELIST M record which either precedes or follows the NAMELIST M which defines radiation surface number 5. Radiation heat transfer is calculated using the temperature at the center of the material region's partition which is adjacent to the indicated surface element; since the theory of radiation heat transfer is based on surface temperatures, the user should use a small partition size (DR) in the material region which is immediately adjacent to the radiation surface, so that the temperature in the material partition which is adjacent to the surface is a close approximation to the surface temperature. The additional input required when radiation surfaces are used is described in Sec. 5.4.

If multiple material regions are used to define a thermal structure prototype, then the regions

are separated from each other by a gap which is void of material. The nature of this gap is described by input variables HGAP and SGAP in NAMELIST M; the values specified in these two variables apply to the gap between the material region defined by the current NAMELIST M and the nextmost inner material region which is defined by the next NAMELIST M for this thermal structure prototype. Heat transfer across the gap is by convection, using the heat transfer coefficient U_{gap} . (Since the finite difference formulation in the code defines thermal structure temperatures at the centers of material partitions, this heat transfer near the interface of two material regions is composed of conduction through a half thickness of the last partition in the outer material region, followed by convection (using U_{gap}) across the gap, followed by conduction through a half thickness of the first partition of the nextmost inner material region.) Input variable HGAP specifies the value of U_{gap} ; note that this is not a heat transfer coefficient correlation index number (which was the case for IHT in NAMELIST F) but is the actual value of a constant heat transfer coefficient. The value input for HGAP must not be zero. To indicate perfect thermal contact between adjacent material regions, enter an extremely large value for HGAP (e.g. 10^{20}). The value input using SGAP is the thickness of the gap spacing; its value is only relevant for cylindrical and spherical thermal structures, for which it is used to calculate volumes of and interfacial areas between partitions properly as a function of radius. Values for HGAP and SGAP should not be specified in the final (or only) NAMELIST M of a thermal structure prototype, since there is no material region further inward to which they would form a connection.

5.3 Thermal Structure Location Input Block

Thermal structures are located in the fluid grid system by specifying which fluid cells interact with which surface of each thermal structure prototype. It is, in fact, these specifications that divide a thermal structure prototype into thermal structure elements and define the surface elements of the thermal structures. These locations are specified using the Thermal Structure Location input block. The record structure of this block is shown in Table 5.5.

Table 5.5 Thermal Structure Location Input Block Record Structure

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier; may be: 'OUT ': the cells specified interact with the outside surface of prototype "NUM"; 'IN ': the cells specified interact with the inside surface of prototype "NUM"; 'END ': terminates this input block; record positions 5-32 should be blank.	NAMelist F preceded NAMelist M. NAMelist F followed NAMelist M.
5-8	I4	NUM	thermal structure prototype number (the 'N' in NAMelist T).	$1 \leq \text{NUM} \leq \text{PNSTRU}$
9-12	I4	IB	beginning I-range index.	$1 \leq \text{IB} \leq \text{IE}$
13-16	I4	IE	ending I-range index.	$\text{IB} \leq \text{IE} \leq \text{IMAX}$
17-20	I4	JB	beginning J-range index.	$1 \leq \text{JB} \leq \text{JE}$
21-24	I4	JE	ending J-range index.	$\text{JB} \leq \text{JE} \leq \text{JMAX}$
25-28	I4	KB	beginning K-range index.	$1 \leq \text{KB} \leq \text{KE}$
29-32	I4	KE	ending K-range index.	$\text{KB} \leq \text{KE} \leq \text{KMAX}$
33-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. This input block is present if, and only if, ISTRUC=2 and NEWTS=1, which is only permitted if ISTATE=0 or =2.
2. These records result in allowing fluid cells in spatial positions (I,J,K) within the three-dimensional region delimited by $\text{IB} \leq \text{I} \leq \text{IE}$, $\text{JB} \leq \text{J} \leq \text{JE}$, and $\text{KB} \leq \text{K} \leq \text{KE}$ within the system volume to interact thermally with the "NAME" surface of thermal structure prototype number "NUM". In connection with note 3(a) below, the user needs to be aware of the sequence in which cells in the specified region will be referenced by the code; using implied-DO notation, this is given by (((cell(I,J,K), I=IB,IE),J=JB,JE),K=KB,KE).
3. Use as many 'OUT ' and 'IN ' records as needed to identify interactions between fluid cells and the surfaces of thermal structures. The order of the records is irrelevant, except (a) if fluid cells interact with both surfaces of a thermal structure, then cells on the outside are paired with cells on the inside to form thermal structure elements in the order in which cells are specified and (b) the 'END ' record only occurs once and is last.
4. The indices (I,J,K) refer to cell-center grid points; the cells must actually exist (i.e. contain fluid).

Table 5.5 Thermal Structure Location Input Block Record Structure (Cont'd)

5. If both surfaces of a thermal structure interact with fluid cells: (a) the cells as paired under note 3(a) become thermally coupled by the thermal structure; (b) the total number of cells specified in 'OUT' records must be the same as the total number of cells specified in 'IN' records.
6. A given set of cell indices (I,J,K) should not be specified more than once for any thermal structure surface unless the real intention is for there to be multiple paths of interaction.
7. When using 'full' coupling between thermal structures and fluid cells (IHTWAL(99)=2), a fluid cell (I1,J1,K1) (a) must not be coupled to itself by a thermal structure; (b) must not be coupled to any other fluid cell (I2,J2,K2) by more than one thermal structure; (c) may be coupled to none, any, or all of its six nearest neighbor adjacent cells (i.e. cells $(I1 \pm 1, J1, K1)$, $(I1, J1 \pm 1, K1)$, and $(I1, J1, K1 \pm 1)$); (d) may be coupled to no more than NTSCUP nonadjacent cells (e.g. cells $(I1 + 4, J1, K1)$, $(I1, J1 - 3, K1)$, and/or $(I1, J1, K1 + 5)$), where the value of NTSCUP is set by a PARAMETER statement in the source coding at installation time.

This input block consists of 'OUT' and 'IN' records which specify which fluid cells are adjacent to the outer and inner surface, respectively, of each thermal structure prototype. If, and only if, fluid cells interact with the outside surface of a thermal structure prototype (which was indicated by there being a NAMELIST F between the NAMELIST T and the (first) NAMELIST M), then one or more 'OUT' records must be present for this prototype. If, and only if, fluid cells interact with the inside surface of a thermal structure prototype (which was indicated by there being a NAMELIST F after the last (or only) NAMELIST M), then one or more 'IN' records must be present for this prototype. If fluid cells interact with both surfaces of a thermal structure, then both 'OUT' and 'IN' records must be present; it does not matter whether 'OUT' records precede or follow 'IN' records.

The value entered for NUM identifies the thermal structure prototype index number (i.e. the N in NAMELIST T) with which to associate the fluid cells being specified. Appropriate 'OUT' and/or 'IN' records must be specified for each thermal structure prototype which is defined in the Thermal Structure Prototype input block. The value of NUM does not have to be strictly nondecreasing as you move from the beginning to the end of this input block nor does the value of NUM have to follow the sequence in which thermal structure prototype index numbers were defined in the Thermal Structure Prototype input block; however, one of these two orderings is usually most convenient for the user. For example, records for thermal

structure number 5 may appear before, after, or intermixed with the records for thermal structure number 10.

Each 'OUT' and 'IN' record specifies a region of one or more fluid cells which interacts with the indicated surface of the thermal structure prototype. The order in which cells are specified using these records is only important if fluid interacts with both surfaces of a thermal structure, which is discussed in the next paragraph. The same cell should not be specified more than once in the 'OUT' or 'IN' records for thermal structure prototype number NUM unless the actual intent is for there to be more than one thermal connection between that fluid cell and thermal structure prototype number NUM; such multiple connections should normally be treated using $|RODFR| > 1$ instead.

When fluid cells interact with both surfaces of a thermal structure prototype, the order in which cells are specified using 'OUT' and 'IN' records is important. The cells specified in all 'OUT' records for a given prototype are grouped into an "OUT list" in the order in which the 'OUT' records occur in this input block and the cells specified in all 'IN' records for the prototype are grouped into an "IN list" in the order in which the 'IN' records occur in this input block. Cells in the "OUT list" are then paired with cells in the "IN list", the first cell in the "OUT list" being paired with the first cell in the "IN list", the second cell in the "OUT list" being paired with the second cell in the "IN list", etc. The same total number of cells must be specified on both sides of the thermal structure, and the cells in each pair become thermally linked to each other by the thermal structure element. For example, if thermal structure number 10 is used to represent a heat-conducting horizontal plate separating fluid cells in the region $(2 \leq I \leq 4, J \leq 5, K = 8)$ from fluid cells in the region $(2 \leq I \leq 4, J \leq 5, K = 9)$, then some of the permissible ways of specifying the 'IN' and 'OUT' records are shown in Table 5.6. Each fluid cell specified by the 'IN' records is not restricted to referencing a cell location which has a lower index value than the cell location to which it is paired by the 'OUT' records for a thermal structure prototype; for example, if fluid cell (I,J,K) interacts with the inside surface of a thermal structure element, then it might be paired by the user with either fluid cell (I,J,K-1) or (I,J,K+1) which interacts with the outside surface of this thermal structure element. Normally, the fluid cells in each pair are adjacent to each other (e.g. fluid cell (I,J,K) on the

inner surface and fluid cell (I,J,K+1) on the outer surface), as is the case for the example in Table 5.6. Adjacency, however, is not a requirement; the user may have fluid cell (I,J,K) interact with the inner surface and be paired with fluid cell (I,J,K+5) which interacts with the outer surface, if that corresponds to the physical situation.

Table 5.6 Some of the Acceptable Forms for Thermal Structure Location
Input Block Records for Slab-Type Thermal
Structure Number 10 Separating Fluid Cells
($2 \leq I \leq 4, J=5, K=8$) from Fluid Cells ($2 \leq I \leq 4, J=5, K=9$)

Method	Record Position							
	111111111122222222223333...778 123456789012345678901234567890123...890							
#1	IN	10	2	2	5	5	8	8
	OUT	10	2	2	5	5	9	9
	IN	10	3	3	5	5	8	8
	OUT	10	3	3	5	5	9	9
	IN	10	4	4	5	5	8	8
	OUT	10	4	4	5	5	9	9
#2	IN	10	4	4	5	5	8	8
	OUT	10	4	4	5	5	9	9
	IN	10	3	3	5	5	8	8
	OUT	10	3	3	5	5	9	9
	IN	10	2	2	5	5	8	8
	OUT	10	2	2	5	5	9	9
#3	IN	10	2	2	5	5	8	8
	IN	10	3	3	5	5	8	8
	IN	10	4	4	5	5	8	8
	OUT	10	2	2	5	5	9	9
	OUT	10	3	3	5	5	9	9
	OUT	10	4	4	5	5	9	9
#4	OUT	10	2	2	5	5	9	9
	OUT	10	3	3	5	5	9	9
	OUT	10	4	4	5	5	9	9
	IN	10	2	2	5	5	8	8
	IN	10	3	3	5	5	8	8
	IN	10	4	4	5	5	8	8
#5	IN	10	2	2	5	5	8	8
	IN	10	3	3	5	5	8	8
	IN	10	4	4	5	5	8	8
	OUT	10	2	4	5	5	9	9
#6	IN	10	2	4	5	5	8	8
	OUT	10	2	4	5	5	9	9

The specification of which fluid cells interact with which surfaces of each thermal structure prototype divides the thermal structure into thermal structure elements. This process may also be viewed as assembling individual thermal structure elements to form a particular physical structure. (Recall that heat transfer within a thermal structure element is in one dimension in the direction between the inside and outside surfaces and there is no heat transfer from one thermal structure element to another except by radiation at one of these two surfaces; thus, the thermal structure elements assembled to form a large physical structure are thermally isolated from each other unless the user forms a connection via the definition of radiation surfaces.) The way in which this process works depends of the basic geometric shape and the orientation, as specified in input variable IXYZ, and is discussed in the next three paragraphs for the three basic shapes. The meanings of "in" and "through" as used in the next three paragraphs will be discussed in the fourth paragraph below.

For a slab, a planar thermal structure element whose unit surface-normal vector is aligned with the fluid coordinate axis given by D (the rightmost digit of IXYZ) and whose surface area is given by RODFR is placed "in" each of the specified cells. The specification of fluid cells allows the user to assemble a large plate which crosses "through" many cells and/or to form a series of parallel plates each of which is "in" a different fluid cell. Using a horizontal slab (IXYZ=13) as an example, specifying a range of fluid cell locations ($2 \leq I \leq 4, 5 \leq J \leq 7, K=8$) for the inside surface and ($2 \leq I \leq 4, 5 \leq J \leq 7, K=9$) for the outside surface replicates this element to form a large slab which crosses "through" 9 pairs of cells in axial locations $K=8$ and $K=9$; on the other hand, specifying a range of fluid cell locations ($I=2, J=5, 1 \leq K \leq 9$) for the inside surface and ($I=2, J=5, 2 \leq K \leq 10$) for the outside surface will replicate this element to form a series of 9 parallel slabs, each of which is "in" a pair of axially adjacent fluid cells (i.e. cells $(2,5,K)$ and $(2,5,K+1)$ for $K=1,2,\dots,9$).

For a cylinder, a thermal structure element having a circular cross-sectional area and whose axis is aligned with the fluid coordinate axis D is placed "in" each of the specified fluid cells; the height of this cylindrical thermal structure element is the fluid mesh spacing in direction D. The specification of fluid cells allows the user to assemble a long cylinder which passes "through" many cells and/or to form an array of parallel cylinders "in" many cells. Using a

vertical cylinder ($IXYZ=3$) which is solid as an example, specifying a range of fluid cells ($I=2, J=5, 1 \leq K \leq 9$) for the outside surface replicates the cylindrical element to form a long cylinder crossing "through" one fluid cell at each of the 9 axial locations $K=1$ through $K=9$; specifying a range of fluid cells ($2 \leq I \leq 3, 5 \leq J \leq 6, 1 \leq K \leq 9$) for the outside surface would replicate the long cylinder in the previous example in 4 different spatial positions in the I-J plane. In either case, the basic cylinder being replicated in each cell is single cylinder ($|RODFR| = 1$), multiple cylinders ($|RODFR| > 1$), or a fractional part of a cylinder ($|RODFR| < 1$).

For a sphere, a thermal structure element having a spherical shape is placed "in" each of the specified fluid cells. Unlike slabs and cylinders, the intersection of the fluid mesh does not divide a sphere into thermal structure elements; the basic shape replicated in each cell is a single complete sphere ($|RODFR| = 1$), multiple complete spheres ($|RODFR| > 1$), or a fractional part of a complete sphere ($|RODFR| < 1$). The specification of fluid cells merely replicates this basic sphere "in" many cells. Thus, the outside radius of the sphere should (conceptually) be sufficiently small that the sphere(s) will actually fit in each of the specified fluid cells. A sphere which is much larger than a single fluid cell must be represented using several different thermal structure prototypes, each of which has a $RODFR (< 1)$ to specify what fraction of the total sphere is "in" each fluid cell; this is a rather tedious process for all but the simplest fluid grid geometries. The orientation index (D) for a sphere is only used in selecting the meaning of the spatial index (m) to be used in the power shape array QK; it has absolutely no bearing on the perceived orientation of a spherical section within any particular fluid cell.

The words "in" and "through" as used in the preceding three paragraphs have special meaning in connection with defining thermal structure prototype locations. The most important concept is that, from the code's perspective, a thermal structure element is not really located within any specific fluid cell; the 'OUT' and 'IN' records merely specify which fluid cells interact with which surfaces of which thermal structure elements. A second related point is that the specification of 'OUT' and 'IN' records to connect thermal structure elements to fluid cells does not automatically alter the volume porosity of any fluid cell or the directional surface porosity of any fluid cell face; it is the user's responsibility to make these adjustments. These

adjustments of volume porosity (γ_v) and directional surface porosities (γ_x , γ_y , γ_z) are specified using 'AL ', 'ALX ', 'ALY ', and 'ALZ ' records, respectively, in the Internal Cell Initialization input block (cf. Sec. 3.5). The next two paragraphs illustrate these points using examples of a slab and a cylinder. (The user is reminded that specification of nonunity volume and surface porosities does not automatically invoke any frictional resistance; if the presence of a thermal structure is to represent an additional resistance term in the momentum equation, then the user must provide input to describe this resistance, using the force structure model described in Sec. 6.)

Consider a heat conducting slab whose unit surface-normal vector is aligned with the z axis; the inside surface of this slab interacts with fluid cells in spatial locations ($2 \leq I \leq 4, 5 \leq J \leq 7, K=8$), and the outside surface interacts with fluid cells in spatial locations ($2 \leq I \leq 4, 5 \leq J \leq 7, K=9$), as shown in Fig. 5.9. The user would input RODFR=-1.0 to indicate that the heat transfer area for each of the 9 thermal structure elements of this slab is to be the same as the interfacial area ($\Delta x_i \Delta y_j$) between the inside and outside fluid cells. To allow this thermal structure to represent a blockage to fluid flow between the inside and outside cells, the user would additionally set the z-direction surface porosity to zero for these cell interfaces; this is accomplished by inputting a value of 0.0 using 'ALZ ' records in the Internal Cell Initialization input block for spatial locations ($2 \leq I \leq 4, 5 \leq J \leq 7, K=8$). For the slab to displace fluid from the fluid cell volumes, the user would additionally need to set the volume porosity of cells adjacent to the inside and/or outside surfaces to an appropriate value; this is accomplished using the 'AL ' records in the Internal Cell Initialization input block. If the slab thickness (which is the sum of (NP*DR+SGAP) over all NAMELIST M's used to define this thermal structure prototype) is L and the slab volume is to be split equally between the fluid cells on the two sides, then the user would enter a value of $(1-(L/2)/\Delta z_k)$ using 'AL ' records for cell locations ($2 \leq I \leq 4, 5 \leq J \leq 7, 8 \leq K \leq 9$). Similarly, the thickness of this slab partially blocks the flow area in the other two directions; the user would enter a value of $(1-(L/2)/\Delta z_k)$ using 'ALX ' records for spatial locations ($1 \leq I \leq 4, 5 \leq J \leq 7, 8 \leq K \leq 9$) and the same value using 'ALY ' records for spatial locations ($2 \leq I \leq 4, 4 \leq J \leq 7, 8 \leq K \leq 9$). (Note that the low-end value in the I range for the 'ALX ' records is 1, not 2, and in the J range for the 'ALY ' records is 4, not 5, to properly modify the surface porosity at the edge of the slab.)

Fig. 5.9 Projections of Grid Geometry Showing Placements of Slab- and Cylinder-Type Thermal Structures

Consider a heat conducting solid cylinder whose axis is aligned with the z axis; the outside surface of this cylinder interacts with fluid cells in spatial locations ($4 \leq I \leq 5, 2 \leq J \leq 3, 4 \leq K \leq 7$), as shown in Fig. 5.9. The user would input $RODFR=0.25$ to indicate that one quarter of the circular cross section is to be the basic thermal structure repeated within each fluid cell. To allow this cylinder to represent a blockage to fluid flow, the user would additionally need to set the x-, y-, and z-direction surface porosities in the vicinity of this cylinder using 'ALX ', 'ALY ', and 'ALZ ' records in the Internal Cell Initialization input block. The user would specify a value of $(1-OUTR/\Delta y_i)$ using 'ALX ' records for spatial locations ($I=4, 2 \leq J \leq 3, 4 \leq K \leq 7$), a value of $(1-OUTR/\Delta x_i)$ using 'ALY ' records for spatial locations ($4 \leq I \leq 5, J=2, 4 \leq K \leq 7$), and a value of $(1-(\pi*OUTR*OUTR/4)/(\Delta x_i*\Delta y_i))$ using 'ALZ ' records for spatial locations ($4 \leq I \leq 5, 2 \leq J \leq 3, 3 \leq K \leq 7$). (Note that the low-end value in the K range for 'ALZ ' records is 3, not 4, in order to properly modify the fluid flow area at the bottom of the cylinder.) For the cylinder to displace fluid from the fluid cell volumes, the user would additionally need to set the volume porosity of the cells adjacent to the outside surface; the user would specify a value of $(1-(\pi*OUTR*OUTR/4)/(\Delta x_i*\Delta y_i))$ for spatial locations ($4 \leq I \leq 5, 2 \leq J \leq 3, 4 \leq K \leq 7$) using 'AL ' records in the Internal Cell Initialization input block.

Many 'OUT ' and 'IN ' records may be needed to specify the fluid-structure interactions for each thermal structure prototype. After these interactions have been specified for all thermal structure prototypes, an 'END ' record must be present to signify the end of the Thermal Structure Location input block.

5.4 Additional User Input

Additional input for the thermal structure model is provided by the user in NAMELISTs GEOM, DATA, and STRUCT. The specific input variables in each of these NAMELISTs are shown in Tables 5.7 through 5.9. In addition to these variables, the user needs to define structure-fluid heat transfer correlations (cf. Sec. 9.3), define properties for structure materials (cf. Sec. 9.2), define transient functions (cf. Sec. 4.1.4), modify volume and directional surface porosities (cf. Sec. 3.5), and define frictional resistances (cf. Sec. 6).

Use of the thermal structure model is optional and is indicated by a nonzero value for ISTRUC in NAMELIST GEOM (cf. Table 5.7). If thermal structures are to be used, the complete set of input which defines them (i.e. Thermal Structure Prototype and Location input

Table 5.7 Thermal Structure Variables in NAMELIST GEOM

Variable	Meaning/Value	Restrictions
ISTRUC	<p>thermal structure use indicator; may be:</p> <p>0: no thermal structures are used;</p> <p>1: thermal structures are used but no new input is included in this computer run;</p> <p>2: thermal structures are used and are completely defined by input in this computer run.</p>	<p>ISTATE > 0</p> <p>ISTATE = 0 or 2 NEWTS = 1</p>

blocks and the variables in NAMELISTs GEOM, DATA, and STRUCT) must be present in the card-image input file at the beginning of a steady-state calculation (ISTATE=0); the user must input ISTRUC=2 in NAMELIST GEOM and NEWTS=1 in NAMELIST DATA (cf. Table 5.8) in order for NAMELIST STRUCT and the Thermal Structure Prototype and Location input blocks to be read. Thermal structure prototype definitions and locations may not be changed (or augmented or diminished) during the continuation of a steady-state (ISTATE=1) or transient (ISTATE=3) calculation; therefore, in the card-image input file, the user must specify ISTRUC=1 and NEWTS=0 and must omit NAMELIST STRUCT and the Thermal Structure Prototype and Location input blocks. If necessary, thermal structures may be redefined when beginning a transient calculation from a steady-state calculation (ISTATE=2); the user specifies ISTRUC=2 and NEWTS=1 and includes NAMELIST STRUCT and the Thermal Structure Prototype and Location input blocks; the information in the Thermal Structure Prototype and Location input blocks must cover all thermal structure prototypes and locations, not just those which the user wishes to change. Although redefinition is allowed at the beginning of a transient, this is not recommended, since the initial temperature distribution will not be consistent with steady state; the preferred approach is to rerun the steady-state part of the calculation from ISTATE=0 with the revised thermal structure input. If no thermal structures are used in a calculation, the user must specify ISTRUC=0 and NEWTS=0 and omit NAMELIST STRUCT and the Thermal Structure Prototype and Location input blocks.

Table 5.8 Thermal Structure Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
NEWTS	<p>indicator of thermal structure definition input in this computer run: denotes presence of NAMELIST STRUCT and the Thermal Structure Prototype and Location input blocks in this file; may be:</p> <p>0: none included;</p> <p>1: all included.</p>	<p>ISTATE=0 or 2 ISTRUC=2</p>
IHTWAL(99)	<p>coupling of thermal structure and fluid energy equations; affects the calculation of the derivative of the heat transfer rate from the surfaces of a thermal structure to the adjacent fluid cell; may be:</p> <p>0: original method; surface temperature held constant in forming derivative;</p> <p>1: limited; surface temperature is a function of adjacent cell's fluid temperature in forming derivative;</p> <p>2: full; surface temperature is a function of temperature in fluid cells on both sides of structure in forming derivative.</p> <p>(Caution: if there is a boundary surface number 99, IHTWAL(99) has dual meaning; see Table 4.6.)</p>	<p>See note 7 in Table 5.5.</p> <p>NSURF=99</p>
QK(m)	<p>QK(m): spatial shape multiplier [-] for heat sources; meaning of m depends on IXYZ in NAMELIST T.</p> <p>(Caution: QK is used as shape multiplier for duct walls, internal fluid, thermal structure, and hexagonal fuel bundle heat sources.)</p>	<p>$1 \leq m \leq \text{IMAX or JMAX or KMAX}$ $Q \neq 0$</p>
NTOTS(NTS)	<p>nf: transient function number to be used as heat source multiplier for thermal structure prototype NTS; these numbers override the NT field specified in the NTS-th NAMELIST T in the Thermal Structure Prototype input block in this run or a prior run in this calculational sequence.</p>	<p>ISTATE=0 or 2 $0 < \text{NTOTS} \leq 25$ $1 \leq \text{NTS} \leq \text{PNSTRU}$</p>
TVAL(NP)	<p>x: values of the independent variable for transient functions.</p>	<p>See Table 4.4.</p>

Table 5.8 Thermal Structure Variables in NAMELIST DATA (Cont'd)

Variable	Meaning/Value	Restrictions
FVAL(NP)	$f(x)$: values of the dependent variable for transient functions.	See Table 4.4.
NEND(NTF)	number of data points used to define transient function number NTF.	See Table 4.4.
IFRAD	radiation heat transfer model usage indicator; may be: 0: model is not used; 2: model is used.	ISTATE=0 or 2
	The values of the following 4 variables are only processed if the radiation heat transfer model is used.	IFRAD>0
NRAD	total number of radiating surfaces in the system volume.	ISTATE=0 or 2 $1 \leq \text{NRAD} \leq \text{PNRAD}$
EMITS(L)	ϵ_L : emissivity [-] of radiation surface L.	$1 \leq L \leq \text{NRAD}$ $0 \leq \text{EMITS}(L) \leq 1$
FVIEW(L,M)	$F_{L,M}$: view factor: fraction of radiation leaving surface L which arrives at surface M; input values must obey reciprocity and conservation laws.	$1 \leq L \leq \text{PNRAD}$ $1 \leq M \leq \text{PNRAD}$ $0 \leq \text{FVIEW}(M,L) \leq 1$
OMR	ϕ_r : underrelaxation factor for advancement of radiation heat flux across a time step.	$0 < \text{OMR} \leq 1$

The value of input variable IHTWAL(99) in NAMELIST DATA selects the degree of coupling employed when forming the derivative of the structure-fluid heat transfer rate. There are three choices: original, limited, and full which are selected by values of 0, 1, and 2, respectively, for IHTWAL(99). The numerics employed in each of the three techniques are described in Sec. 6.3.5 of the Equations and Numerics report (Ref. 1). All three techniques give the same structure-fluid heat transfer rate at the end of a converged steady-state calculation and at the end of each converged time step during a transient calculation; they differ in the amount of computational work done per time step and will give different answers at the end of

each time step but the last in a steady-state calculation, since the solution is only fully converged at the end of the final time step. The full coupling technique is the nominal choice recommended; this technique requires the most work per outer iteration but generally requires fewer time steps to converge a steady-state calculation and fewer outer iterations to converge a time step during a transient calculation. Use of the full coupling technique places some restrictions on the ways in which fluid cells may be connected to thermal structures (cf. note 7 in Table 5.5); the two most important of these are (a) a pair of fluid cells may only be coupled by a single thermal structure (i.e. the same cell location indices must not become paired by 'OUT' and 'IN' records in the Thermal Structure Location input block for more than one thermal structure index number NUM) and (b) a fluid cell may be coupled to no more than NTSCUP nonadjacent fluids cells by thermal structures, where the value of NTSCUP is set internally in the code and typically has a value of 2 but this value may have been altered by the person who installed COMMIX at your site (cf. Appendix C.1.1.). The limited and original coupling techniques require less work per outer iteration than with the full coupling technique but generally require more outer iterations than with the full coupling technique to converge a time step during a transient; however, one of these two techniques may actually be more efficient than the full coupling technique when very small time step sizes are used or the thermal coupling between the structure and the fluid cell(s) is somewhat loose (e.g. low heat transfer coefficient). There are several other notes in connection with IHTWAL(99): (a) the user may change calculational techniques during a continuation run at ISTATE > 0; (b) the user must input IHTWAL(99)=0 if there are no thermal structures in a calculation; (c) if there is a 99th boundary surface, then the value of IHTWAL(99) not only selects the thermal-structure-to-fluid coupling technique (for all thermal structures) but also selects the fluid-to-boundary-surface heat transfer coefficient correlation for boundary surface number 99 (cf. Sec. 4.1.3).

Several variables in NAMELIST DATA are connected with heat sources in thermal structures, which are described by the discussion surrounding Eq. 5.1. Values input in array QK are the spatial shape distribution of the heat source; the meaning of the the index m depends on the alignment of the thermal structure. Note that the single array QK is used as a spatial shaping distribution for heat sources in several models within the code. Values may be input using the QK array at any time in the calculational sequence. The NTOTS array may be used

to change the transient function numbers to be associated with the heat sources in thermal structures; the values input using this array are only recognized at the beginning of a steady-state calculation (ISTATE=0) and when beginning a transient calculation from a steady-state calculation (ISTATE=2). A nonzero value in location NTS of array NTOTS will override the value specified using variable NT in NAMELIST T for the NTSth thermal structure prototype definition; the index NTS corresponds to the order in which thermal structure prototypes are defined in the Thermal Structure Prototype input block, rather than the thermal structure number which was assigned to the prototype using variable N in NAMELIST T. The NTOTS array is particularly useful for changing a heat source from being constant during a steady-state calculation to being a function of time during a transient calculation; the transient function number is input in this way to avoid having to reread (and redefine) all thermal structures to make this change. Any zero values input using array NTOTS will be ignored, rather than causing a heat source to be changed from being a function of time to being constant. If any heat sources are a function of time, the user must describe the transient functions using input variables TVAL, FVAL, and NEND in NAMELIST DATA; these input variables are described in Sec. 4.1.4.

The final group of input in NAMELIST DATA is for radiation heat transfer among surfaces of thermal structures. Use of the radiation heat transfer model is optional, and the user must make the decision at the time when thermal structure prototypes are being defined. The values input using variable KRAD in the NAMELIST M's in the Thermal Structure Prototype input block indicate which surface(s) of which thermal structure prototype(s) is to be radiation surface number KRAD; the values input for KRAD are only processed if the value of IFRAD in NAMELIST DATA is nonzero (i.e. 2). If the radiation heat transfer model is to be used, there are four additional variables which the user must input using NAMELIST DATA: NRAD, EMITS, FVIEW, and OMR; values must be specified for these variables at the time when thermal structure prototypes are defined; revised values of EMITS, FVIEW, and OMR may be input at subsequent times in the calculational sequence. The value of NRAD indicates the number of radiation surfaces, which is the maximum value entered using KRAD in any of the NAMELIST M records in the Thermal Structure Prototype input block. Array EMITS is used to specify the emissivity (ϵ) of each radiation surface; the values entered must be positive and must not exceed 1.0.

Array FVIEW is used to supply the view factors ($F_{L,M}$); the value entered in array element FVIEW(L,M) is the fraction of radiant energy flux leaving radiation surface L which arrives at radiation surface M. View factors are a function of the size, shape, and relative locations between radiation surfaces. The rather large range of possible geometric configurations for radiation surfaces precludes the automatic calculation of view factors within COMMIX. Formulas for calculating view factors may be found in various heat transfer texts and reference handbooks; a particularly extensive compilation of view factors is given in Ref. 8. A given radiation surface may undergo radiant energy exchange with one or more other radiation surfaces. For any pair of radiation surfaces L and M, the user must specify a value for the view factor from surface L to surface M ($F_{L,M}$) and a value for the view factor from surface M to surface L ($F_{M,L}$); see rule (d) below. View factors are specified for the entire radiation surface (which is composed of one or more thermal structure surface elements), rather than for each surface element of the radiation surface. The value of each view factor lies in the inclusive range from 0.0 (none of the radiant energy leaving radiation surface L reaches radiation surface M) to 1.0 (all of the radiant energy leaving radiation surface L reaches radiation surface M); since a concave surface can see (at least a part of) itself, $F_{L,L}$ is not necessarily 0.0. The user must observe several rules when inputting array FVIEW: (a) the array is two-dimensional and has a DIMENSION declaration of FVIEW(PNRAD,PNRAD), where the value of PNRAD is set internally in the code and typically has a value of 99 but this value may have been altered by the person who installed COMMIX at your site (cf. Appendix C.1.1.); (b) when reading a two-dimensional array from a NAMELIST in FORTRAN, the first subscript increases the fastest; therefore, the values input for this array are processed in the following order (using implied-DO notation): (FVIEW(L,M), L=1,PNRAD), M=1,PNRAD); (c) the user must fill locations with 0.0 for $NRAD < L \leq PNRAD$ and for $NRAD < M \leq PNRAD$; (d) the input values must satisfy the reciprocity relation,

$$FVIEW(L,M) * (area_L) = FVIEW(M,L) * (area_M) , \quad (5.2)$$

where $area_i$ is the surface area of radiation surface i; (e) the input values should represent a conservation of radiant energy exchange, i.e.

$$\sum_{M=1}^{PNRAD} FVIEW(L,M) = 1.0 \quad (5.3)$$

for each radiation surface L.

The final piece of radiation heat transfer information in NAMELIST DATA is variable OMR. The value of OMR is the (under) relaxation factor (Ω_r) used in advancing the radiation heat fluxes from one time step to the next (cf. Eq. 6.48 of Ref. 1). The value of OMR needs to be as large as possible (but not exceeding 1.0) so the values of the radiation heat fluxes are in synch with the rest of the calculation; however, values which are too large will cause numerical instability, since the coupling between the radiation heat transfer model and the rest of the code is fairly explicit in time. The choice is case dependent. A relatively small value (e.g. 0.05 or less) is generally needed for OMR during the early stages of a steady-state calculation; larger values of OMR should be used during the later stages of a steady-state calculation and during a transient calculation.

The final piece of thermal structure input is NAMELIST STRUCT. As shown in Table 5.9, there is only one variable to be specified using this NAMELIST. The value of ITSBUG controls printing of the "Thermal Structure Control Block", which is a table showing index offsets for storage of thermal structure variables and echoing some of the information input in the Thermal Structure Prototype input block. This table is only useful during debugging processes.

Table 5.9 Thermal Structure Variables in NAMELIST STRUCT

Variable	Meaning/Value	Restrictions
ITSBUG	selects printing of Thermal Structure Control Block, showing index offsets in storage; may be: 0: not printed; 1: printed.	

Notes:

1. This input block is present if, and only if, ISTRUC=2 and NEWTS=1, which is only permitted if ISTATE=0 or =2.

6. FORCE STRUCTURES

The presence of solid structures in or surrounding the fluid field has three effects: geometrical, thermal, and frictional. The geometrical effects are included by the way in which the user defines boundary surfaces and nonunity values for volume and surface porosities (cf. Sec. 3). The thermal effects are included by the way in which the user defines energy boundary conditions (cf. Sec. 4.1.3) and thermal structures (cf. Sec. 5). The frictional resistances associated with these solids are governed by the force structure model, described in this section. Any flow restriction that causes the flow field to deviate from a continuum field should be represented as a force structure.

Use of the force structure model is optional, and, thus, the associated input is optional. The features of the force structure model are summarized below in Sec. 6.1. The input associated with the force structure model is described in Sec. 6.2.

6.1 Review of Model Features

The presence of solid structures in or near the fluid domain influences the fluid flow by providing increased flow resistance. These effects are incorporated in COMMIX using the distributed flow resistance terms, R_j , in the momentum equations (cf. Table 2.1 of Ref. 1). Specification of nonzero values for these terms is not automatic except for (a) the wire-wrap and pin effects under the hexagonal fuel assembly option (cf. Sec. 10) and (b) the effects of zero-slip rigid-wall boundaries under one of the nonconstant turbulence models (which enter the momentum equations through V_j , as discussed in Sec. 8). Instead, the user is left with the flexibility (and responsibility) to specify which locations in the fluid grid have a distributed resistance, the direction of the resistance, and the parameters describing the resistance.

In the literature, the pressure drop ΔP due to the presence of a structure in the flow field is expressed in numerous forms, such as

$$\Delta P = - f \frac{L}{D} \frac{1}{2} \rho |u| u, \quad (6.1a)$$

$$\Delta P = - 4f' \frac{L}{D} \frac{1}{2} \rho |u| u, \quad (6.1b)$$

$$\Delta P = - C_D \frac{1}{2} \rho |u| u, \quad (6.1c)$$

or

$$\Delta P = - K \frac{1}{2} \rho |u| u, \quad (6.1d)$$

where L and D are characteristic lengths, ρ is the fluid density, and u is the fluid velocity. The quantities f , f' , C_D , and K are given various names in the literature, such as the Fanning friction factor, the Darcy-Weisbach friction factor, the Moody friction factor, the drag coefficient, and the loss coefficient. Whatever the name, these multipliers are a function of the flow geometry and, usually, the local fluid flow rate. The numerical values and/or functional forms of these multipliers may be found in fluid dynamics texts and handbooks (e.g. Refs. 9 and 10) for many common shapes. The user may need to refer to other sources for more complex geometries; for example, several forms of this multiplier for nuclear reactor fuel assemblies containing wire-wrapped fuel rods are given in Refs. 11, 12, and 13. A useful compilation of these multipliers was given in the report on a previous version of COMMIX (Appendix C of Ref. 6).

The approach in COMMIX is to represent the various forms of Eq. 6.1 using a general form for a distributed resistance (R_j), which, in terms of a pressure change (ΔP) per length in coordinate direction j (L_j), is given by

$$R_j \equiv - \frac{\Delta P}{L_j} = C_{fn} f_m \frac{1}{D_{h1n}} \rho |u_j| u_j \quad (6.2)$$

where C_{fn} is a dimensionless force-multiplier constant, D_{h1n} is a characteristic length, f_m is a dimensionless friction factor, ρ is the fluid density, and u_j is the fluid velocity in direction j . C_{fn} , D_{h1n} , and the friction factor correlation index number m are input by the user for each force structure number n .

The user has three ways to define each friction factor correlation: use the general form, use a form defined in the friction factor library, or add a new form to the friction factor library. The general friction factor expression provided in COMMIX allows the friction factor to be a two-range (essentially, laminar and turbulent) function of the Reynolds number and is given by

$$f_m = \begin{cases} C_{aFtm} Re^{C_{bFtm}} + C_{cFtm} , & Re \leq Re_{tm} \\ C_{aFtm} Re^{C_{bFtm}} + C_{cFtm} , & Re > Re_{tm} \end{cases} \quad (6.3a)$$

$$(6.3b)$$

where the Reynolds number is defined as

$$Re = \frac{\rho D_{h2n} |u_j|}{\mu_t} \quad (6.4)$$

and the six dimensionless constants C_{dFem} (for $d=a,b,c$ and $e=\ell,t$) and transition Reynolds number Re_{tm} must be input by the user for each of the m different correlation functions to be used; up to 20 such functions may be defined. The user must also input the characteristic length D_{h2n} to be used in the Reynolds number for each of the n distributed resistances. Note that the fluid viscosity μ_t used in Reynolds number definition (Eq. 6.4) is the laminar (molecular) component, not the effective (laminar plus turbulent) viscosity.

If the friction correlation can not be adequately represented by the above general form, the user has two options. First, the user should check to see if an appropriate form is already in the code's friction factor library. The correlations currently in this library are shown in Table 6.1. Second, if an applicable correlation is not in the library, the user may request that the desired correlation be added to the library by the person who maintains the code. (The COMMIX friction factor library is part of SUBROUTINE FORCES; additional details are in the Programmer's Guide (Ref. 14).)

The spatial positioning of frictional resistances in the system volume is accomplished by the user specifying which momentum control volumes are to use a specific one of the n force structure prototypes. This is a two-part specification of (a) the type of momentum control volume (i.e. x - (or r -), y - (or θ -), or z -direction) and (b) the specific location in the fluid grid

Table 6.1 Friction Factor Correlations in Library

# Use	Correlation
81 IFR Fuel Assembly	$Re \leq 1000:$ $f = 84/Re$ $Re > 1000:$ $f = [1.080 + 0.49625*(1000/Re)^2 + 0.01481*(1000/Re)^4]$ $*0.25169/Re^{0.22609}$ Effectively, $D_{hin} = D_{h2n} = 2.784*10^{-3}$ m.
82 IFR Blanket Assembly	All Re: $f = 110/Re + 0.48/Re^{0.25}$ Effectively, $D_{hin} = D_{h2n} = 2.774*10^{-3}$ m.
83 IFR Shield Assembly	Same as given by #82.
84 0.8636 m Diameter Pipe	$Re \leq 1000:$ $f = f_t(Re)$ $1000 < Re < 2000:$ $f = (1-\chi)*f_t(Re) + \chi*f_i(Re)$ $Re \geq 2000:$ $f = f_i(Re)$ where $f_t(Re) = 64/Re$ $f_i(Re) = 0.0055*[1 + (2.3 + 10^6/Re^{0.3333})]$ $\chi = [(Re-1000)/1000]^{0.5}$ Effectively, $D_{hin} = D_{h2n} = 0.8636$ m.
85 IHX	Same as given by #84.
86 DRACS	Same as given by #84.
87 Stopped Pump Rotor	$Q < 0.02:$ $f = 4.03414/ u_j $ $Q \geq 0.02:$ $f = 0.0573583*\rho^{0.9}/ u_j ^{0.1}$ where $Q = \rho* u_j /5640.5677$ Effectively, $D_{hin} = 0.682871$ m and D_{h2n} is not used.

Table 6.1 Friction Factor Correlations in Library (Cont'd)

# Use	Correlation
90 CRBR Fuel Assembly	$Re \leq 400:$ $f = f_t(Re)$ $400 < Re < 5000:$ $f = (1-\chi)^{0.5} f_t(Re) + \chi^{0.5} f_i(Re)$ $Re \geq 5000:$ $f = f_i(Re)$ where $f_t(Re) = 81.7/Re$ $f_i(Re) = 0.48/Re^{0.25}$ $\chi = (Re-400)/4600$ Effectively, $D_{hin} = D_{h2n} = 3.25 \cdot 10^{-3}$ m.
91 CRBR Blanket Assembly	$Re \leq 400:$ $f = f_t(Re)$ $400 < Re < 5000:$ $f = (1-\chi)^{0.5} f_t(Re) + \chi^{0.5} f_i(Re)$ $Re \geq 5000:$ $f = f_i(Re)$ where $f_t(Re) = 99/Re$ $f_i(Re) = 0.48/Re^{0.25}$ $\chi = (Re-400)/4600$ Effectively, $D_{hin} = D_{h2n} = 3.39 \cdot 10^{-3}$ m.
92 Direct Reactor Heat Exchanger	All Re: $f = \{0.171 + 0.012 \cdot (P/D) - 0.07 \cdot \exp[-50 \cdot (P/D - 1)]\} / Re^{0.22}$ where $P/D = 1.84$ Effectively, $D_{hin} = D_{h2n} = 0.10546$ m.
93 CRBR Chimney	$Re \leq 1200:$ $f = f_t(Re)$ $1200 < Re < 4000:$ $f = (1-\chi)^{0.5} f_t(Re) + \chi^{0.5} f_i(Re)$ $Re \geq 4000:$ $f = f_i(Re)$ where $f_t(Re) = 64/Re$ $f_i(Re) = 0.3164/Re^{0.25}$ $\chi = (Re-1200)/2800$ Effectively, $D_{hin} = D_{h2n} = 0.127$ m.
94 FFTF Fuel Assembly	$Re \leq 1000:$ $f = 84/Re$ $Re > 1000:$ $f = [1 + 0.1746 \cdot (1000/Re)^2 + 0.0745 \cdot (1000/Re)^4] \cdot 1.075 / f_c^2$ where $f_c = -0.8686 \cdot \ln(2.51 \cdot f_c / Re)$ Effectively, $D_{hin} = D_{h2n} = 3.95 \cdot 10^{-3}$ m.

Table 6.1 Friction Factor Correlations in Library (Cont'd)

# Use	Correlation
95 CRBR Control Assembly	$Re \leq 400:$ $f = f_t(Re)$ $400 < Re < 5000:$ $f = (1-\chi)^{0.5} f_t(Re) + \chi^{0.5} f_t(Re)$ $Re \geq 5000:$ $f = f_t(Re)$ where $f_t(Re) = 60.68/Re$ $f_t(Re) = 0.48/Re^{0.25}$ $\chi = (Re-400)/4600$ Effectively, $D_{hin} = D_{h2n} = 3.48 \times 10^{-3} \text{ m.}$
In all of the above, $Re = \rho * D_{h2n} * u_j / \mu_t$ is the Reynolds number, ρ is the fluid density, D_{hin} and D_{h2n} are characteristic lengths, u_j is the velocity component in direction j , μ_t is the laminar (molecular) fluid viscosity, and all quantities are in SI units.	

system. Each momentum control volume may be assigned a frictional resistance described by, at most, one force structure prototype. The direction of the resistance is the same as the direction of the momentum control volume and has a sign which is opposite to the sign of the velocity associated with that momentum control volume. The resulting distributed flow resistance is applied over the length of the corresponding momentum control volume, which is offset by a half a cell from the main control volume. For example, Fig. 6.1 shows a x-direction momentum control volume which spans the space from grid point (i,j,k) to grid point $(i+1,j,k)$ and is composed of half of the two adjacent main control volumes which are centered at grid points (i,j,k) and $(i+1,j,k)$. The length of the momentum control volume in Fig. 6.1 is $(\Delta x_i + \Delta x_{i+1})/2$, which becomes the L_j term in Eq. 6.2. The velocity to be used in Eq. 6.2 and 6.4 is the x-direction velocity u at the interface between main control volumes as shown in Fig. 6.1; recall that velocities are always defined on the faces of the main control volumes in the staggered-grid finite-difference formulation in COMMIX. The y- (or θ -) and z-momentum control volumes are similarly offset by half a cell in the j - and k -directions, respectively, from the main control volume.

Fig. 6.1 Illustration of Staggered Grid Concept, Showing Difference Between Main and Momentum Control Volumes

The user needs to be aware of the correspondence between the physical geometry being modeled, the geometry as defined to COMMIX, and the basis of friction factor correlations. Using as an example the case of a sudden change in flow area shown in Fig. 6.2a, the pressure drop associated with this might be presented in the literature as a function of the velocity in the small area section, e.g.,

$$\Delta P = - K_1 \frac{1}{2} \rho |u_1| u_1, \quad (6.5a)$$

or as a function of the velocity in the large area section, e.g.,

$$\Delta P = - K_2 \frac{1}{2} \rho |u_2| u_2, \quad (6.5b)$$

Fig. 6.2 Flow Through a Sudden Enlargement: (a) Basic Geometry, (b) Placement in Grid with Reference Velocity u_1 , and (c) Placement in Grid with Reference Velocity u_2

where the subscripts 1 and 2 refer to the small and large area sections, respectively. The various flow areas in any problem are set by a combination of (cf. Sec. 3) the grid geometry (i.e. Δx (or Δr), Δy (or $r\Delta\theta$), and Δz), the volume porosities (γ_v) which are defined for main control volumes, and the surface porosities (i.e. γ_x (or γ_r), γ_y (or γ_θ), and γ_z) which are defined on the faces of the main control volumes. The various porosities alter the meaning of the velocity, since it is the product of density times grid area times surface porosity times velocity (e.g. $\rho \Delta y \Delta z \gamma_x u$) which is used to define the mass flow rate across a surface. Returning to the example, the problem geometry defined in Fig. 6.2b would have to use a friction factor correlation based on the velocity in the small area section (which corresponds to Eq. 6.5a), and the problem geometry defined in Fig. 6.2c would have to use a friction factor correlation based on the velocity in the large area section (which corresponds to Eq. 6.5b). If one finds one type of these two types of correlations in the literature but needs the other type, the conversion from one type to the other is can usually be made; for the example in Fig. 6.2, continuity of fluid flow dictates that $K_1 = K_2*(A_1/A_2)^2$.

6.2 User Input

The force structure input is in three sections of the card-image input file: NAMELISTs GEOM and DATA and the Force Structure Location input block. This input is similar in concept to the input for the thermal structure model (cf. Sec. 5), in that the user defines a set of force structure prototypes and then specifies in which momentum control volume(s) each force structure prototype should be placed. The frictional resistance associated with a given physical object may need to be described using more than one force structure prototype; for example, the frictional resistance associated with flow parallel to the axis of a group of rods has a different form than the frictional resistance to flow in the directions which are transverse to the axis. On the other hand, the same force structure prototype may be used to represent the presence of several similarly shaped physical objects at various locations in the fluid flow field. Many force structure prototypes may be needed to represent the frictional effects of all of the physical objects in a complex system geometry.

Use of the force structure model is optional and is indicated by the value input for the variable **NFORCE** in **NAMelist GEOM**, as shown in Table 6.2. A calculation having no force structures will have **NFORCE**=0; the remainder of the input described in this section must be omitted, except that the user should input **NEWFOR**=0 in **NAMelist DATA**. Use of the force structure model is indicated by **NFORCE**>0. Each force structure prototype is identified by an index number, which is the subscript "n" in Eqs. 6.2 and 6.4; these index numbers must be in the range from 1 through 99 (inclusive). Although not a requirement, the user will find it most convenient to assign index numbers in a contiguous sequence beginning with 1. The value of **NFORCE** in **NAMelist GEOM** indicates the maximum value of the force structure prototype index numbers n used in the problem. (The value of **NFORCE** is the number of force structure prototypes if the index numbers are assigned contiguously beginning with 1.) The value of this variable must be input at the beginning of a steady-state calculation (i.e. **ISTATE**=0); the value may be changed when beginning a transient calculation from a steady-state calculation (i.e. **ISTATE**=2); the value must not be changed (and need not be present in the card-image input file) when continuing a steady-state or a transient calculation in the same calculational mode (i.e. **ISTATE**=1 or 3).

Table 6.2 Force Structure Variables in **NAMelist GEOM**

Variable	Meaning/Value	Restrictions
NFORCE	max(n) : maximum force structure prototype index number.	ISTATE =0 or 2 $0 \leq \text{NFORCE} \leq 99$

The remainder of the force structure prototype definition input is contained in **NAMelist DATA**, as shown in Table 6.3. For each force structure prototype n, the user must input a value for 4 array elements: **ICORR(n)**, **FORCEF(n)**, **CLENTN(n)**, and **REYLEN(n)**.

Table 6.3 Force Structure Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
NEWFOR	Indicator of force structure definition input in this computer run: denotes presence of Force Structure Location input block in this file; may be: 0: none included; 1: all included:	ISTATE=0 or 2 NFORCE>0
	A force structure imposes a pressure gradient, $\Delta P/L_j = -[C_{fn} * f_{m(n)}(Re)/D_{hin}] * \rho * u_j * u_j,$ in the momentum control volumes indicated in the Force Structure Location input block. The next 4 variables must be input for each force structure prototype number n.	$1 \leq n \leq \text{NFORCE}$
ICORR(n)	m: type of friction factor correlation; may be: 1-20: selects general form (i.e. $f(Re) = C_{sf} * Re^{C_{sf}} + C_{cf}$ where the Reynolds number is $Re = \rho * D_{kin} * u_j / \mu_t$) to which the user supplies (A B C)CORR(L T) and REYTRN; 50-99: selects alternate form from friction factor library; may be (cf. (Table 6.1): 81: IFR fuel assembly; 82: IFR blanket assembly; 83: IFR shield assembly (same as #82); 84: 0.8636 m i.d. pipe; 85: IHX (same as #84); 86: DRACS (same as #84); 87: stoppered centrifugal pump rotor; 90: CRBR fuel assembly; 91: CRBR blanket assembly; 92: direct reactor heat exchanger; 93: CRBR chimney assembly; 94: FFTF fuel assembly; 95: CRBR control assembly. (Contact person who installed COMMIX at your site for other library forms.)	
FORCEF(n)	C_{fn} : force coefficient multiplier in $\Delta P/L_j$ expression.	
CLENTH(n)	D_{hin} : length [m] used in $\Delta P/L_j$ expression; if $CLENTH < 0$, then this length will be set internally to the length of the momentum control volume to which the force is applied.	$CLENTH \neq 0$ Ignored if ICORR=81-87,90-95
REYLEN(n)	D_{kin} : length [m] used in calculating the Reynolds number to be used in the friction correlation.	$REYLEN > 0$ Ignored if ICORR=81-87,90-95
	The values of the following 8 variables must be input if the general friction form is used.	$1 \leq \text{ICORR} \leq 20$
NCORR	maximum of the ICORR(n) values which are less than or equal to 20; the following 7 variables must be input for each correlation referenced by a value of ICORR ≤ 20 :	$1 \leq \text{NCORR} \leq 20$
REYTRN(m)	Re_{tm} : Reynolds number at which to switch between laminar ($Re \leq Re_{tm}$) and turbulent ($Re > Re_{tm}$) friction correlations.	$1 \leq m \leq \text{NCORR}$

Table 6.3 Force Structure Variables in NAMELIST DATA (Cont'd)

Variable	Meaning/Value	Restrictions
ACORRL(m)	$C_{aF/m}$: Reynolds number multiplier in laminar friction.	$1 \leq m \leq \text{NCORR}$
BCORRL(m)	$C_{bF/m}$: Reynolds number exponent in laminar friction.	$1 \leq m \leq \text{NCORR}$
CCORRL(m)	$C_{cF/m}$: additive term in laminar friction.	$1 \leq m \leq \text{NCORR}$
ACORRT(m)	$C_{aF/m}$: Reynolds number multiplier in turbulent friction.	$1 \leq m \leq \text{NCORR}$
BCORRT(m)	$C_{bF/m}$: Reynolds number exponent in turbulent friction.	$1 \leq m \leq \text{NCORR}$
CCORRT(m)	$C_{cF/m}$: additive term in turbulent friction.	$1 \leq m \leq \text{NCORR}$

The values input in array ICORR specify which friction factor correlation number m is to be used for each force structure prototype. A value between 1 and 20 (inclusive) in ICORR selects the general friction factor correlation form given by Eq. 6.3; the constants to be provided for each one of these forms are described in the third paragraph below; although not a requirement, the user will find it most convenient to assign friction factor correlation numbers in this range in a contiguous manner beginning with 1. A value of ICORR between 50 and 99 (inclusive) selects a friction factor correlation from the friction factor library in the code. Currently, this library defines forms (cf. Table 6.1) for correlation numbers 81 through 87 and 90 through 95; note that each of these correlations is for a very specific geometry. Additional friction factor correlation forms having index numbers in the range 50 through 99 may be included in the library in the future. The same friction factor correlation number m may be used for more than one force structure prototype. The values entered in the ICORR array do not have to form a strictly increasing sequence.

The values input in array FORCEF are the force coefficients C_m in Eq. 6.2. This coefficient may be used to account for a number of different effects; two common uses are cited here. First, a typical value for FORCEF is 0.5, since many pressure-drop correlations involve a multiple of the term $0.5 \rho |u| u$ but there is no 0.5 factor automatically included in the form given by Eq. 6.2. Second, the user may need to increase or decrease FORCEF from a base value to account for the length of a force structure being longer or shorter than the momentum control volume; this is necessary if the ends of a physical object align with the boundaries of

main control volumes, since the resistances described by force structures are applied to momentum control volumes, which are offset by half a cell from the main control volumes.

The values input in arrays CLENTH and REYLEN are, respectively, the characteristic lengths D_{h1n} , appearing in Eq. 6.2, and D_{h2n} , used in Eq. 6.4 to calculate the Reynolds number to be used in the friction factor correlation. The values to be input depend on the geometry and the friction factor correlation; in many cases, the same value will be input for both variables; for example, for flow inside of a duct, the value would typically be the hydraulic diameter or $4 \times (\text{flow area}) / (\text{wetted perimeter})$. The values input for CLENTH and REYLEN are both ignored for the predefined friction factor correlations currently in the friction factor library (i.e. for ICORR values in the range 80 through 87 and 90 through 95). If the user inputs a negative value for CLENTH, then the value of D_{h1n} will be set internally by the code to the length of the momentum control volume in which the force structure is located; this effectively introduces a pressure drop of the form $f \rho |u| u$ instead of the more general form $f(L/D) \rho |u| u$.

Each different value of ICORR in the range from 1 through 20 selects a different set of constants to be used in the general friction factor form given by Eq. 6.3. The value to be input for variable NCORR is the maximum of the values entered in array ICORR which are less than or equal to 20. (If the friction factor correlation index numbers which are less than or equal to 20 are used in a contiguous sequence beginning with 1, then the value of NCORR is the number of such correlations.) For each different value of $m = \text{ICORR}(n)$ which is less than or equal to 20, the user must enter values for the three constants in Eq. 6.3a (where the value of $\text{ACORRL}(m)$ is C_{aFtm} , $\text{BCORRL}(m)$ is C_{bFtm} , and $\text{CCORRL}(m)$ is C_{cFtm}), the three constants in Eq. 6.3b (where the value of $\text{ACORRT}(m)$ is C_{aFtm} , $\text{BCORRT}(m)$ is C_{bFtm} , and $\text{CCORRT}(m)$ is C_{cFtm}), and the transition Reynolds number (Re_{tm} , which is given by the value of $\text{REYTRN}(m)$). Although the friction factor forms given by Eq. 6.3a and 6.3b may be viewed as corresponding to, respectively, the laminar and turbulent flow regimes for which the transition from one regime to the other may not be smooth as presented in the literature, the user is encouraged to input the constants and a transition Reynolds number for Eq. 6.3 in such a way that the friction factor correlations for the two regimes actually intersect at the transition Reynolds number in order to avoid numerical instabilities in the code.

The above-mentioned force structure definition variables in NAMELIST DATA must be input at the beginning of a steady-state calculation (i.e. ISTATE=0). These variables need not be present in the card-image file for computer runs which are continuations of the initial run (i.e. ISTATE>0). The values of these variables may be changed at a later time in the calculational sequence by inputting new values; however, there is seldom a need to alter the values of these variables during a calculational sequence.

The final force structure input variable in NAMELIST DATA is NEWFOR. An input value of 1 indicates that the Force Structure Location input block is present in the card-image file for this computer run; this is only allowed at the beginning of a steady-state calculation (i.e. ISTATE=0) and when beginning a transient calculation from a steady-state calculation (i.e. ISTATE=2). An input value of 0 indicates that the Force Structure Location input block is not present in the card-image file for this computer run; this is the only valid choice when continuing a steady-state or a transient calculation in the same mode (i.e. ISTATE=1 or 3, respectively); a value of 0 is also required at the beginning of a steady-state calculation which has no force structures (i.e. NFORCE=0) and when beginning a transient calculation from a steady-state calculation with no changes in force structure locations.

The final group of force structure input is the Force Structure Location input block. The purpose of this input block is to associate the frictional resistance described by each force structure prototype with one or more momentum control volumes. The presence of this input block in the card-image input file is indicated by NEWFOR=1 in NAMELIST DATA; therefore, this input block may only be present at the beginning of a steady-state calculation (i.e. ISTATE=0) and when beginning a transient calculation from a steady-state calculation (i.e. ISTATE=2); in either case, this input block must contain location information for all force structure prototypes. (Although the force structure locations are allowed to be respecified at the beginning of a transient calculation, this is not recommended: the steady-state velocity field, which was calculated with the prior force structure locations, may not be consistent with the new force structure locations and, thus, may not present a proper initial condition. The preferred approach is to rerun the steady-state calculation from ISTATE=0 with the correct force structure specifications; then begin the transient calculation from this new steady-state calculation.)

The record structure of the Force Structure Location input block is shown in Table 6.4. This input block consists of 'XFOR', 'YFOR', and 'ZFOR' records which specify, respectively, which x- (or r-), y- (or θ -) and z-direction momentum control volumes are to contain a distributed resistance described by a particular force structure prototype number. The direction of the momentum control volume is also the direction of the resultant resistance. The value entered for NUM identifies the force structure prototype index number (i.e. the subscript n in Eqs. 6.2 and 6.4) which is used to describe the resistance in each of the momentum control volumes being specified. Appropriate 'XFOR', 'YFOR', and/or 'ZFOR' records must be specified for each force structure prototype which is defined in NAMELIST DATA. (If this input block is present at ISTATE=2, then location information must be specified for all force structure prototypes, not just for those which are to be changed from locations specified in a prior run in the calculational sequence.)

Each 'XFOR', 'YFOR', and 'ZFOR' record specifies a region of one or more momentum control volumes within which the force structure prototype is to be located. The meaning of the spatial location indices (I,J,K) depends on the type of record: for 'XFOR' records, this identifies the x- (or r-) momentum control volume which is located between grid points (I,J,K) and (I+1,J,K) (cf. Fig. 6.1); for 'YFOR' records, this identifies the y- (or θ -) momentum control volume which is located between grid points (I,J,K) and (I,J+1,K); for 'ZFOR' records, this identifies the z-momentum control volume located between grid points (I,J,K) and (I,J,K+1). (Recall that grid points are located at the centers of main, not momentum, control volumes.) These location indices should only reference momentum control volumes which exist; recalling that a momentum control volume is composed of half of two adjacent main control volumes, then a momentum control volume exists only if both of the main control volumes of which it is composed also exist. The presence of a boundary surface affects the existence of momentum control volumes: there is no j-direction momentum control volume defined for the space (half of a main control volume) between a boundary surface whose unit surface-normal vector points in direction j and the grid point in the center of the main control volume which is adjacent to the boundary surface and toward which the unit surface-normal vector points. Each momentum control volume may contain, at most, one force structure prototype; multiple occurrences of the same force structure prototype in the same momentum control volume must

Table 6.4 Force Structure Location Input Block Record Structure

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier, indicating the direction of the force; may be: 'XFOR': x (or r) direction; 'YFOR': y (or θ) direction; 'ZFOR': z direction; 'END ': terminates this input block; record positions 5-32 should be blank.	
5-8	I4	NUM	force structure prototype index number.	$1 \leq \text{NUM} \leq \text{NFORCE}$
9-12	I4	IB	beginning I-range index.	$1 \leq \text{IB} \leq \text{IE}$
13-16	I4	IE	ending I-range index.	$\text{IB} \leq \text{IE} \leq \text{IMAX}$
17-20	I4	JB	beginning J-range index.	$1 \leq \text{JB} \leq \text{JE}$
21-24	I4	JE	ending J-range index.	$\text{JB} \leq \text{JE} \leq \text{JMAX}$
25-28	I4	KB	beginning K-range index.	$1 \leq \text{KB} \leq \text{KE}$
29-32	I4	KE	ending K-range index.	$\text{KB} \leq \text{KE} \leq \text{KMAX}$
33-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. This input block is present if, and only if, $\text{NFORCE} > 0$ and $\text{NEWFOR} = 1$, which is only permitted if $\text{ISTATE} = 0$ or $= 2$.
2. These records result in adding a frictional resistance acting in direction "NAME" and described by force structure prototype number "NUM" to the "NAME"-direction momentum control volume at spatial positions (I,J,K) in the three-dimensional region delimited by $\text{IB} \leq \text{I} \leq \text{IE}$, $\text{JB} \leq \text{J} \leq \text{JE}$, and $\text{KB} \leq \text{K} \leq \text{KE}$ within the system volume.
3. The meaning of the indices (I,J,K) depends on the direction of the force. For 'XFOR' records, these indices refer to the x- (or r-) momentum control volume which is located between fluid grid locations (I,J,K) and (I+1,J,K). For 'YFOR' records, these indices refer to the y- (or θ) momentum control volume which is located between fluid grid locations (I,J,K) and (I,J+1,K). For 'ZFOR' records, these indices refer to the z-direction momentum control volume which is located between fluid grid locations (I,J,K) and (I,J,K+1).
4. Use as many 'XFOR', 'YFOR', and 'ZFOR' records as needed to associate force structure prototypes with the proper momentum control volumes. Each momentum control volume may contain, at most, one force structure prototype. The order of the records is irrelevant, except (a) if the same momentum control volume is referenced by more than one record, the final such specification will be used, and (b) the 'END' record occurs only once and is last.
5. Force structures may only be placed in momentum control volumes which exist. For a momentum control volume to exist, the two main control volumes of which it is composed must both exist (i.e. contain fluid) and must not be separated from each other by a boundary surface.

be treated using an appropriately larger value of the force multiplier (as input using variable FORCEF) rather than referencing the same momentum control volume several times in this input block.

As a specific example of the specification of force structure locations, consider a force structure which is to be placed uniformly over the length of four momentum control volumes, as shown in Fig. 6.3. The records in the Force Structure Location input block would specify the four momentum control volume locations shown in the upper part of Fig. 6.3. The lower part of Fig. 6.3 shows the main control volumes associated with these momentum control volumes and is a reminder that the main and momentum control volumes are offset from each other (by half a cell); thus, the four momentum control volumes correspond to three complete main control volumes plus half of two more main control volumes.

Fig. 6.3 Example of Force Structure Location Relative to the Two Types of Control Volumes: (a) Momentum and (b) Main

The order in which 'XFOR', 'YFOR', and 'ZFOR' records occur in the Force Structure Location input block is irrelevant; for example, 'XFOR' records may precede, follow, and be intermixed with 'YFOR' and 'ZFOR' records. The value of NUM is not required to be strictly increasing as you move from the beginning to the end of this input block; thus, records associated with force structure prototype number 5 may precede, follow, and be intermixed with records associated with force structure prototype number 10. The same force structure prototype may be used in more than one direction's momentum control volume if that corresponds to the physics of the problem (e.g. the crossflow frictional resistance in a regularly spaced array of rods whose axis is aligned with the z direction would be properly characterized using the same force structure prototype for both the x- and y-direction momentum control volumes).

Many 'XFOR', 'YFOR', and 'ZFOR' records may be needed to specify the frictional resistances associated with each force structure prototype. After these resistances have been specified for all force structure prototypes, an 'END ' record must be present to signify the end of the Force Structure Location input block.

This completes the description of the input associated with the force structure model. The user is reminded that the force structure model only describes the frictional aspects of the presence of a solid object in the fluid field. Additional input is generally required to describe the geometrical aspects, using nonunity values of porosities (cf. Sec. 3.5), and the thermal effects, using energy boundary conditions (cf. Sec. 4.1.3) and the thermal structure model (cf. Sec. 5).

7. PUMPS

7.1 Selection of Model

The inclusion of modeling for pumps in a COMMIX calculation is optional and is indicated by the value input for variable IFPUMP in NAMELIST DATA. As shown in Table 7.1, IFPUMP=0 indicates pumps are not being modeled, and IFPUMP > 0 indicates pumps are being modeled. A value must be entered for IFPUMP at the beginning of a steady-state calculation (ISTATE=0); the value of IFPUMP may be changed when beginning a transient calculation from a steady-state calculation (ISTATE=2); however, the value of IFPUMP must not be changed (and, thus, IFPUMP need not be included in NAMELIST DATA) when continuing a steady-state or a transient calculation in the same mode (ISTATE=1 or 3). (When pumps are not modeled, NAMELIST PUMP and the Pump Location input block described below must be omitted from the card-image input file.)

When pumps are being modeled, the value of IFPUMP selects the type of model to be used for all pumps in the calculation. The four choices available are shown in Table 7.1: homologous with full range operation (IFPUMP=3), homologous with reverse rotation precluded (IFPUMP=4), specified speed (IFPUMP=1), and specified pressure increase (IFPUMP=2). These models are described in Sec. 8 of Ref. 1. The end result of each of these pump models is a pressure increase supplied by the pump (ΔP_p), which is applied across the length of the momentum cell in which the pump is located. This pressure increase may be a function of time (t) and, depending on the model, other pump quantities, such as speed (N), motor torque (τ_m), and volumetric flow rate (Q).

The additional input which describes the pumps is provided by the user in two places in the card-image input file: NAMELIST PUMP and the Pump Location input block. These two sets of input may only be present at the beginning of a steady-state calculation (ISTATE=0) and when beginning a transient calculation from a steady-state calculation (ISTATE=2). The input to be included in NAMELIST PUMP depends on the type of pump model selected; this input is described in Sec. 7.2 for the two homologous models, in Sec. 7.3 for the specified-speed

Table 7.1 Pump Model Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
IFPUMP	<p>selects type of model used for all pumps in the calculation; may be:</p> <p>=0: no pumps; omit NAMELIST PUMP and Pump Location input block;</p> <p>>0: pump(s) of type IFPUMP are included; specify additional input in NAMELIST PUMP and Pump Location input block; specific pump types are:</p> <p>=1: user-specified speed;</p> <p>=2: user-specified pressure increase;</p> <p>=3: full range homologous model;</p> <p>=4: homologous model with reverse flow prevented.</p>	<p>ISTATE=0 or 2</p> <p>ISTATE=2</p>
TVAL(NP)	x: values of the independent variable for transient functions.	See Table 4.4.
FVAL(NP)	f(x): values of the dependent variable for transient functions.	See Table 4.4.
NEND(NTF)	number of data points used to define transient function number NTF.	See Table 4.4.

model, and in Sec. 7.4 for the specified-pressure-increase model. The input to be included in the Pump Location input block is the same for all pump models and is described in Sec. 7.5. All of the pump models use transient functions in some manner; these functions are indexed by entries in arrays NTORQM and NTPUMP in NAMELIST PUMP; the transient functions are defined using input arrays TVAL, FVAL, and NEND in NAMELIST DATA, as discussed in Sec. 4.1.4.

Several prefacing remarks are appropriate before proceeding with the detailed input description. Each pump used in a calculation is identified by an index number, which is denoted by "NP" in what follows; the user must assign these index numbers in a contiguous sequence beginning with 1. These are discrete pump index numbers (rather than being "prototype" numbers, as was the case for thermal and force structures). There may be, at most, 5 pumps in a calculation. The input provided in NAMELIST PUMP describes the

characteristics of each pump. The Pump Location input block then specifies the spatial location of each pump. Unless stated otherwise, the user may assume that the input described below is to be provided at the beginning of a steady-state calculation (ISTATE=0); only limited portions of this input may be changed when beginning a transient calculation from a steady-state calculation (ISTATE=2). Both NAMELIST PUMP and the Pump Location input block must be physically present in the card image input file at ISTATE=2 when pumps are being modeled; if no changes are to be made to the pump parameters (as entered previously at ISTATE=0), then NAMELIST PUMP will be empty (cf. Sec. 2) and the Pump Location input block will consist of just the 'END ' terminator record.

7.2 Homologous Model

The homologous pump model is selected by IFPUMP=3 or 4 in NAMELIST DATA. The equations used in COMMIX to implement this model are presented in Sec. 8.2 of Ref. 1. The difference between the two choices is that IFPUMP=3 allows full range operation, whereas IFPUMP=4 prevents reverse shaft rotation. The additional input to be provided by the user in NAMELIST PUMP is shown in Table 7.2; with one exception (CORPT8), the same input variables are used for IFPUMP=3 and IFPUMP=4. These input variables are discussed in the remainder of this section.

Table 7.2 Steady-State Homologous Pump Model (IFPUMP=3 or 4)
Variables in NAMELIST PUMP¹

Variable	Meaning/Value	Restrictions
NPUMP	number of pumps.	$1 \leq \text{NPUMP} \leq 5$
PUMPS(NP)	N: pump speed [rev/min].	$1 \leq \text{NP} \leq \text{NPUMP}$
PUMPSR(NP)	N_R : reference pump speed [rev/min].	$1 \leq \text{NP} \leq \text{NPUMP}$
QFLOWR(NP)	Q_R : reference pump volumetric flow rate [m^3/s]; sign is same convention as for fluid velocity.	$1 \leq \text{NP} \leq \text{NPUMP}$ $\text{QFLOWR} \neq 0$
TORQR(NP)	τ_{hR} : reference torque [$\text{N}\cdot\text{m}$].	$1 \leq \text{NP} \leq \text{NPUMP}$ $\text{TORQR} > 0$
PUMPHR(NP)	H_R : reference pump head [m].	$1 \leq \text{NP} \leq \text{NPUMP}$ $\text{PUMPHR} > 0$

Table 7.2 Steady-State Homologous Pump Model (IFPUMP=3 or 4)
Variables in NAMELIST PUMP¹ (Cont'd)

Variable	Meaning/Value	Restrictions
DENRO	$\rho * g$: (fluid density)*(gravitational acceleration) [kg/m ² /s ²]. (Used to convert from 'head' [m] to 'pressure' [Pa].)	DENRO $\neq$ 0
ICORP(NP)	pump correlation number; may be: = 1: general homologous pump; = 50: CRBR primary pump.	$1 \leq NP \leq NPUMP$
CORPT8(i)	C_{pi} : coefficients in hydraulic torque function WT(X) [-]; input values for either: full range operation; reverse rotation precluded.	ICORP(NP) = 1 IFPUMP = 3 $1 \leq i \leq 9$ IFPUMP = 4 $10 \leq i \leq 18$
CORPH8(i)	C_{hi} : coefficients in head function WH(X) [-].	ICORP(NP) = 1 $1 \leq i \leq 9$
CORPF(i)	C_{fi} : coefficients in frictional torque function WF($\tilde{N}$) [-].	ICORP(NP) = 1 $1 \leq i \leq 9$
CUTPF(i)	$C_{f,ca,i}$: range-transition points (relative speed) used in frictional torque function WF($\tilde{N}$) [-].	ICORP(NP) = 1 $1 \leq i \leq 2$
TORQFR(NP)	τ_m : reference frictional torque [N*m].	ICORP(NP) = 50 $1 \leq NP \leq NPUMP$
CORPT(i)	C_{pi} : coefficients [-] in hydraulic torque function WT($\tilde{Q}/\tilde{N}$).	ICORP(NP) = 50 $2 \leq i \leq 6$ Optional.
IPUBUG	select printing from subroutine PUMPQ (# of lines below is per pump, per outer iteration); may be: = 0: none; ≥ 2 : 3 lines at exit; 1 more line if in linear head vs. flow range; ≥ 3 : 2 lines after speed calculation; ≥ 4 : 18 lines during evaluation of hydraulic torque and head; ≥ 5 : 1 or more lines during integration of motor torque.	IPTRAN = 1 ICORP(NP) = 1 IPTRAN = 1

Notes:

1. This input block is present if, and only if, IFPUMP > 0 and ISTATE = 0 or 2.

The value of the variable NPUMP is the number of pumps in the calculation. The value input must be at least 1 but must not exceed 5. For each pump index number NP ($1 \leq NP \leq NPUMP$), the user must provide several quantities: the actual speed (N) using array element PUMPS(NP), the reference speed (N_R) using array element PUMPSR(NP), the reference volumetric flow rate (Q_R) using array element QFLOWR(NP), the reference torque (τ_{hR}) using array element TORQR(NP), the reference head (H_R) using array element PUMPHR(NP), and the pump characteristic correlation number using array element ICORP(NP). Note that the units for speed (both in the input and internally in the code) are [revolution/minute], not [revolution/second]. In many cases, the same value should be entered for the actual speed and the reference speed for a steady-state calculation, but this depends on how the user constructs the characteristic functions; however, the user must not enter a value for the actual speed during a transient calculation -- this speed is calculated internally by the code. The user has two choices for the characteristic correlation number: ICORP=1 or ICORP=50, which are discussed in the next three paragraphs; this choice determines the manner in which the characteristic curves for the pump are defined; these characteristic curves describe the relationships between speed, flow rate, torque, and head for a pump operating at steady state.

The general form of the characteristic correlations is selected by specifying ICORP=1; this is the normal user choice. The pump characteristic curves are composed of a set of three relationships, one each for hydraulic torque, frictional torque, and head, which have certain general forms to which the user supplies a set of constants. In particular, the hydraulic torque (τ_h) is given by

$$\tau_h = \tau_{hR} (\tilde{N}^2 + \tilde{Q}^2) * WT(X), \quad (7.1)$$

where

$$\tilde{N} \equiv N/N_R, \quad (7.2)$$

$$\tilde{Q} \equiv Q/Q_R, \quad (7.3)$$

and

$$X \equiv \pi + \tan^{-1} (\tilde{Q}/\tilde{N}). \quad (7.4)$$

The code utilizes the signs of both $\tilde{Q}$ and $\tilde{N}$ as shown in Table 7.3, when evaluating Eq. 7.4 so that X can identify any point in the $\tilde{Q}$ - $\tilde{N}$ plane. The function $WT(X)$ depends on the pump model selection; when using the full range operation option (IFPUMP=3), the function is given by

$$WT(X) = \sum_{i=1}^9 C_{\tau,i} X^{i-1} \quad (7.5a)$$

and the user supplies the values of $C_{\tau,i}$ using array elements CORPT8(i) for $i=1,2,\dots,9$; when using the option which precludes reverse rotation (IFPUMP=4, which restricts X to the range $\pi/2 \leq X \leq 3\pi/2$), the function is given by

$$WT(X) = \sum_{i=1}^9 C_{\tau,i+9} X^{i-1} \quad (7.5b)$$

and the values of $C_{\tau,i+9}$ are input using array elements CORPT8(i+9) for $i=1,2,\dots,9$. The frictional torque τ_f is given by

$$\tau_f = \tau_{hR} * WF(\tilde{N}), \quad (7.6)$$

Table 7.3 Quadrant Assignments for X (Eq. 7.4)

$\tilde{Q}$	$\tilde{N}$	X	Operating Mode
<0	<0	$0 < X < \pi/2$	reverse speed
<0	>0	$\pi/2 < X < \pi$	dissipation
>0	>0	$\pi < X < 3\pi/2$	normal
>0	<0	$3\pi/2 < X < 2\pi$	reverse speed dissipation

where

$$WF(\tilde{N}) = \begin{cases} \sum_{i=1}^3 C_{f,i} \tilde{N}^{i-1} & \text{for } \tilde{N} \leq C_{f,cut,1} \\ \sum_{i=1}^3 C_{f,i+3} \tilde{N}^{i-1} & \text{for } C_{f,cut,1} < \tilde{N} \leq C_{f,cut,2} \\ \sum_{i=1}^3 C_{f,i+6} \tilde{N}^{i-1} & \text{for } C_{f,cut,2} < \tilde{N} \end{cases} \quad (7.7a)$$

$$(7.7b)$$

$$(7.7c)$$

and the user supplies values of $C_{f,i}$ using array elements CORPF(i) for $i=1,2,\dots,9$ and the values of $C_{f,cut,i}$ using array elements CUTPF(i) for $i=1,2$. The head (H) is given by

$$H = H_R (\tilde{N}^2 + \tilde{Q}^2) * WH(X), \quad (7.8)$$

with

$$WH(X) = \sum_{i=1}^9 C_{h,i} X^{i-1} \quad (7.9)$$

and the user supplies values of $C_{h,i}$ using array elements CORPH8(i) for $i=1,2,\dots,9$. Note that only one set of constants is entered using arrays CORPT8, CORPF, CUTPF, and CORPH8, and this set is used for all pump index numbers NP having ICORP(NP)=1; on the other hand, the user may enter different torque and head reference values for each pump using array elements TORQR(NP) and PUMPHR(NP).

The torque and head functions defined by Eqs. 7.5 and 7.9 are called the "characteristic curves" of the pump. Ideally, the information entered for the characteristic curves should be for the actual pump being modeled. Homologous pump theory provides a functional means by which to estimate the characteristic curves of a specific pump for which detailed characteristic curves are not available. The theory is based upon the fact that the characteristic curves for different pumps have similar shapes if the pumps are geometrically and dynamically similar. This allows data taken for one specific pump to be used to characterize a class of similar pumps.

The similarity condition is satisfied when the specific speed, defined by

$$N_s = \frac{N Q^{0.5}}{H^{0.75}}, \quad (7.10)$$

is similar for two pumps. Note that, by convention, the specific speed is not dimensionless. Examples of the head and torque functions are shown in Fig. 7.1 for $N_s=35$, where N is in [revolution/minute], Q is in [m^3/s], and H is in [m]. (This corresponds to $N_s=1800$ when Q is in [gallon/minute] and H is in [foot].) The symbols are the data as given in Ref. 15; the lines are the fits to these data from Ref. 16 (and are the default curves in the code for IFPUMP=3).

Fig. 7.1 Default Characteristic Head (WH) and Torque (WT) Functions Used in the Homologous Pump Model (IFPUMP=3). (Symbols are data for $N_s=35$ (or 1800 -- see text) from Ref. 15 and lines are fits to these data from Ref. 16.)

A specific set of pump characteristic curves is selected by ICORP=50. These curves were developed (Ref. 17) specifically for the primary system sodium pumps in the Clinch River Breeder Reactor design (for which $N_s=40$, where Q is in $[m^3/s]$ and H is in $[m]$). The hydraulic torque is computed from

$$\tau_h = \tau_{hR} \tilde{N}^2 * WT(\tilde{Q}/\tilde{N}), \quad (7.11)$$

where

$$\begin{aligned} WT(\tilde{Q}/\tilde{N}) = & 0.6500 + 0.2738 (\tilde{Q}/\tilde{N}) + 0.2512 (\tilde{Q}/\tilde{N})^2 \\ & -0.1325 (\tilde{Q}/\tilde{N})^3 + 0.0051 (\tilde{Q}/\tilde{N})^4 \end{aligned} \quad (7.12a)$$

is defined internally in the code and the user supplies the reference hydraulic torque (τ_{hR}) in array element TORQR(NP) for each pump. (The code actually computes the hydraulic torque function using

$$WT(\tilde{Q}/\tilde{N}) = \sum_{i=2}^6 C_{\tau,i} (\tilde{Q}/\tilde{N})^{i-2}; \quad (7.12b)$$

the user may override the coefficients shown in Eq. 7.12a by supplying the values of $C_{\tau,i}$ using array elements CORPT(i) for $i=2,3,\dots,6$; the values entered using array CORPT are used for all pump index numbers NP which have ICORP(NP)=50.) The frictional torque is computed from

$$\tau_f = \tau_{fR} * WF(\tilde{N}), \quad (7.13)$$

where the function

$$WF(\tilde{N}) = \begin{cases} 0.01 - 73.13 \tilde{N}^2 & \text{for } \tilde{N} \leq 0.01 \\ 0.00268 + 0.07 \tilde{N}^2 & \text{for } 0.01 < \tilde{N} \leq 0.268 \\ 0.00383 + 0.01071 \tilde{N} + 0.01406 \tilde{N}^2 & \text{for } 0.268 < \tilde{N} \end{cases} \quad \begin{matrix} (7.14a) \\ (7.14b) \\ (7.14c) \end{matrix}$$

is defined internally in the code and the user provides the reference frictional torque (τ_R) in array element TORQFR(NP) for each pump. The head is computed from

$$H = H_R \tilde{N}^2 * WH(\tilde{Q}/\tilde{N}), \quad (7.15)$$

where the function

$$WH(\tilde{Q}/\tilde{N}) = \begin{cases} 1.2655 - 0.0377 (\tilde{Q}/\tilde{N}) + 0.1079 (\tilde{Q}/\tilde{N})^2 \\ -0.3892 (\tilde{Q}/\tilde{N})^3 + 0.0971 (\tilde{Q}/\tilde{N})^4 & \text{for } (\tilde{Q}/\tilde{N}) \leq 2.053 \\ 2.371 - 1.155 (\tilde{Q}/\tilde{N}) & \text{for } (\tilde{Q}/\tilde{N}) > 2.053 \end{cases} \quad (7.16a)$$

$$(7.16b)$$

is defined internally in the code and the user provides the reference head H_R in array element PUMPHR(NP) for each pump.

The head [m], as calculated from Eq. 7.8 or 7.15, is converted to an equivalent pressure change [Pa] using

$$\Delta P_p = (\rho * g) * H, \quad (7.17)$$

where the value of the quantity ($\rho * g$) (i.e density times gravitational acceleration) is input by the user using variable DENRO. (Note that DENRO is not an array; the same value is used for all pumps in a case.) The value of input variable IPUBUG controls the amount of print output generated by the subroutine which calculates pump quantities during the case; a nominal choice is IPUBUG=2; selecting IPUBUG=0 may result in too little information about the pumps, and IPUBUG > 2 is for model debugging purposes.

Additional input, consisting of the seven variables shown in Table 7.4, is required for a transient calculation. This input should be provided when beginning a transient calculation from a steady-state calculation (i.e. ISTATE=2). (With the exception of TORQM, this information may be included in the card-image input file at the beginning of the steady-state calculation (i.e. ISTATE=0); then, if unchanged, the values do not need to be respecified at the beginning of the transient calculation.) The remainder of this section describes the additional input needed for a transient calculation.

Table 7.4 Transient Homologous Pump Model (IFPUMP=3 or 4) Variables in
NAMELIST PUMP¹

Variable	Meaning/Value	Restrictions ²
NTORQM(NP)	nf: transient function number used to compute relative motor torque. (If input NTORQM(NP)=0, then must input NTPUMP(NP)>0; see next item.)	$1 \leq \text{NTORQM} \leq 25$ $\text{NTPUMP(NP)}=0$
NTPUMP(NP)	nf: transient function number used to compute relative pump speed. (If input NTPUMP(NP)=0, then must input NTORQM(NP)>0; see prior item.)	$1 \leq \text{NTPUMP} \leq 25$ $\text{NTORQM(NP)}=0$
PINER(NP)	I_p : moment of inertia of rotating pump elements [$\text{kg}\cdot\text{m}^2$].	$\text{NTORQM(NP)} > 0$ $\text{PINER(NP)} > 0$
IPTRAN	method of integrating motor torque function; may be =0: approximate; or =1: exact.	$\text{NTORQM} > 0$
TORQM(NP)	τ_{mr} : reference motor torque [$\text{N}\cdot\text{m}$]. (Optional; if not input, will be set to steady-state value of ($\tau_h + \tau_f$).)	$\text{NTORQM(NP)} > 0$ $\text{TORQM(NP)} > 0$
NCUT	$\tilde{N}_{cut}$: value of relative speed below which head vs. flow relationship is forced to be linear [-]. (NCUT is a floating-point quantity.)	$\text{ICORP}=1$
QCUT	$\tilde{Q}_{cut}$: value of relative flow below which head vs. flow relationship is forced to be linear [-].	$\text{ICORP}=1$ $\text{QCUT} > 0$

Notes:

1. This input block is present if, and only if, IFPUMP>0 and ISTATE=0 or 2. These variables are only used for ISTATE $\geq$ 2.
2. Where indicated, input values for each pump, NP=1,2,...,NPUMP.

The values input in array elements NTORQM(NP) and NTPUMP(NP) determine the manner in which the speed is calculated for pump number NP. The user must input either (a) NTORQM(NP)>0 and NTPUMP(NP)=0 or (b) NTPUMP(NP)>0 and NTORQM(NP)=0. (If the user inputs NTORQM(NP)>0 and NTPUMP(NP)>0 for the same pump index number NP, then this will be interpreted by the code as selecting case (a) above. User input of NTORQM(NP)=0 and NTPUMP(NP)=0 for the same pump index number NP is not allowed.) The two speed calculation options are described in the next two paragraphs.

Under the option selected by $\text{NTORQM}(\text{NP}) > 0$, the speed of each pump is computed by integrating the differential equation governing rotational motion, which is given by

$$(120\pi) I_p \frac{dN}{dt} = \tau_m - \tau_h - \tau_f, \quad (7.18)$$

where t is time. (The factor of 120π in Eq. 7.18 converts the speed from [revolution/minute] to [radian/second].) The hydraulic and frictional torques appearing on the right-hand side of this equation are computed from Eqs. 7.1 and 7.6 when $\text{ICORP}(\text{NP})=1$ and from Eqs. 7.11 and 7.13 when $\text{ICORP}(\text{NP})=50$. The quantity I_p is the moment of inertia and is input using array element $\text{PINER}(\text{NP})$ for each pump index number NP . The motor torque (τ_m) is assumed to be a function of time and is computed from

$$\tau_m = \tau_{mR} * f_{nf}(t). \quad (7.19)$$

The reference motor torque (τ_{mR}) may be specified by the user using array element $\text{TORQM}(\text{NP})$; if not specified, it will be set internally by the code to the steady-state value of the sum of the hydraulic and frictional torques, or

$$\tau_{mR} = (\tau_h + \tau_f)_s. \quad (7.20)$$

The term $f_{nf}(t)$ is transient function number nf ; the value corresponding to the subscript nf is input using array element $\text{NTORQM}(\text{NP})$. If the motor torque for pump number NP is to be constant during a transient, the user must input a nonzero transient function number in array element $\text{NTORQM}(\text{NP})$ and define this transient function as having a constant value (e.g. 1.0) for all time, rather than inputting $\text{NTORQM}(\text{NP})=0$. The value input in variable IPTRAN selects the method to be used by the code in integrating the motor torque term across a time step (i.e., from t^{n-1} to t^n); when $\text{IPTRAN}=0$, the integral is approximated by $\tau_{mR} * f_{nf}(t^n) * (t^n - t^{n-1})$; when $\text{IPTRAN}=1$, the analytical form corresponding to the exact integral of the transient function is used. (There is little run-time penalty paid for selecting $\text{IPTRAN}=1$, since the transient function is represented internally in the code by a series of third-order polynomial fits to the data points supplied by the user.) When using the model which precludes reverse shaft rotation ($\text{IFPUMP}=4$), if the integration of Eq. 7.18 results in a negative speed, the speed will be set to zero by the code. After the motor torque has decreased to zero and the rotational inertia has

dissipated, the pump becomes an impedance to flow (creating a pressure drop), operating as a turbine being driven by the bouyant head and other mechanisms in the system.

Under the option selected by $NTPUMP(NP) > 0$, the speed is assumed to be a known function of time, given by

$$N = N_x * f_{nf}(t), \quad (7.21)$$

rather than being computed from Eq. 7.18. The term $f_{nf}(t)$ is transient function number nf ; the value corresponding to the subscript nf is input using array element $NTPUMP(NP)$. If the speed for pump number NP is to be constant during a transient, the user must input a nonzero transient function number in array element $NTPUMP(NP)$ and define this transient function as having a constant value (e.g. 1.0) for all time, rather than inputting $NTPUMP(NP)=0$. (Although not explicitly needed, the hydraulic and frictional torques will be computed using Eqs. 7.1 and 7.6 ($ICORP(NP)=1$) or Eqs. 7.11 and 7.13 ($ICORP(NP)=50$). An equivalent motor torque is not computed.)

Regardless of the option used to compute speed during a transient, the head will be computed from the speed and flow using the same relationship used during the steady-state calculation (i.e., Eq. 7.8 for $ICORP(NP)=1$ or Eq. 7.15 for $ICORP(NP)=50$) but with one new provision. Provision has been made for the user to force the head versus flow relationship to be linear at low speed and flow. The following discussion only applies to pumps which are described using the general form of the characteristic correlations (i.e., for which $ICORP(NP)=1$). During the first time step in which the relative speed and relative volumetric flow rate for any pump satisfy both of the following relationships:

$$\tilde{N}_{NP} < \tilde{N}_{cut} \quad (7.22)$$

and

$$\tilde{Q}_{NP} < \tilde{Q}_{cut}, \quad (7.23)$$

the current values of the relative head and the relative volumetric flow rate for that pump are

saved by the code as $\tilde{H}_{test}$ and $\tilde{Q}_{test}$, respectively. During the remainder of the transient, any pump which satisfies the speed and flow relationships given by Eqs. 7.22 and 7.23 will have its head computed using

$$H_{NP} = H_{R,NP} \left(\frac{\tilde{H}_{test}}{\tilde{Q}_{test}} \right) \tilde{Q}_{HP} \quad (7.24)$$

instead of using Eqs. 7.8. The subscript NP has been used in Eqs. 7.22 through 7.24 to emphasize which quantities are a function of the pump index number; note that $\tilde{N}_{cut}$, $\tilde{Q}_{cut}$, $\tilde{Q}_{test}$, and $\tilde{H}_{test}$ are not functions of the pump index number NP. The user provides the values of $\tilde{N}_{cut}$ and $\tilde{Q}_{cut}$ using input variables NCUT and QCUT, respectively, in NAMELIST PUMP. If the user wishes to avoid use of the linear head versus flow rate relationship, then input large negative values for NCUT and QCUT. The user must exercise some caution in selecting values for NCUT and QCUT when more than one pump in a calculation has ICORP(NP)=1.

7.3 Specified Speed Model

The specified-speed pump model is selected by IFPUMP=1 in NAMELIST DATA. The equations used in COMMIX to implement this model are presented in Sec. 8.3 of Ref. 1. The additional input to be provided by the user in NAMELIST PUMP is shown in Table 7.5. These input variables are discussed in the remainder of this section.

The value of the variable NPUMP is the number of pumps in the calculation; the value input must be at least 1 but must not exceed 5. For each pump index number NP ($1 \leq NP \leq NPUMP$), the user must provide several quantities: the actual speed (N) using array element PUMPS(NP), the reference speed (N_R) using array element PUMPSR(NP), the characteristic length (L) using array element PUMLEN(NP), and the pump characteristic correlation number using array element ICORP(NP). Note that the units for speed (both in the input and internally in the code)

Table 7.5 Specified-Speed Pump Model (IFPUMP=1) Variables in NAMELIST PUMP¹

Variable	Meaning/Value	Restrictions
NPUMP	number of pumps.	$1 \leq \text{NPUMP} \leq 5$
PUMPS(NP)	N: pump speed [rev/min].	$1 \leq \text{NP} \leq \text{NPUMP}$
PUMPSR(NP)	N_R : reference pump speed [rev/min].	$1 \leq \text{NP} \leq \text{NPUMP}$
PUMLEN(NP)	L: characteristic length [m] used in pump Reynolds number.	$1 \leq \text{NP} \leq \text{NPUMP}$
ICORP(NP)	pump correlation number.	$1 \leq \text{NP} \leq \text{NPUMP}$ $1 \leq \text{ICORP}(\text{NP}) \leq \text{NCORRP}$
NCORRP	number of pump correlations.	$\text{NCORRP} = 1$
ACORP(NC)	a: overall multiplier (reference pressure increase) [Pa] in pressure increase expression.	$1 \leq \text{NC} \leq \text{NCORRP}$
BCORP(NC)	b: multiplier [-] of $\tilde{N} \cdot \tilde{Q}$ in pressure increase expression.	$1 \leq \text{NC} \leq \text{NCORRP}$
CCORP(NC)	c: multiplier [-] of $ \tilde{Q} $ in pressure increase expression.	$1 \leq \text{NC} \leq \text{NCORRP}$
ALPHAL(NC)	d_l : exponent [-] of $ \tilde{Q} $ in pressure increase expression for $\text{Re} \leq \text{Re}_{tr}$.	$1 \leq \text{NC} \leq \text{NCORRP}$
ALPHAT(NC)	d_t : exponent [-] of $ \tilde{Q} $ in pressure increase expression for $\text{Re} > \text{Re}_{tr}$.	$1 \leq \text{NC} \leq \text{NCORRP}$
REYTRP(NC)	Re_{tr} : transition Reynolds number [-] for choosing between d_l and d_t .	$1 \leq \text{NC} \leq \text{NCORRP}$
IPUBUG	select printing from subroutine PUMPQ (# of lines below is per pump, per outer iteration); may be: =0: none; =1: 1 line at exit.	
NTPUMP(NP)	nf: transient function number used to compute relative pump speed.	$1 \leq \text{NTPUMP} \leq 25$ See note 2.
QFLOWR(NP)	reference pump volumetric flow rate [m^3/s]; sign is same convention as for fluid velocity.	$\text{QFLOWR} \neq 0$ See note 2.

Notes:

1. This input block is present if, and only if, IFPUMP>0 and ISTATE=0 or 2.
2. Value is only used during a transient calculation (ISTATE $\geq$ 2). Input values for each pump, NP=1,2,...,NPUMP.

are [revolution/minute], not [revolution/second]. In many cases, the same value should be entered for the actual speed and the reference speed for a steady-state calculation, but this depends on how the user constructs the characteristic function; however, the user must not enter a value for the actual speed during a transient calculation -- this speed is calculated internally by the code.

The pressure increase supplied by the pump is calculated using

$$\Delta P_p = a(\tilde{N}^2 + b\tilde{N}\tilde{Q} + c|\tilde{Q}|^d), \quad (7.25)$$

where the relative speed ($\tilde{N}$) and the relative flow rate ($\tilde{Q}$) are defined by Eqs. 7.2 and 7.3, respectively. The constants a , b , and c are input by the user in array elements ACORP(NC), BCORP(NC), and CCORP(NC), respectively, in NAMELIST PUMP for $NC=1,2,\dots,NCORRP$. The exponent used in Eq. 7.25 is assumed to be a function of flow rate, as measured by the Reynolds number,

$$Re = \frac{\rho L u_n}{\mu_t}, \quad (7.26)$$

and is selected using

$$d = \begin{cases} d_t & \text{for } Re \leq Re_{tr} \\ d_l & \text{for } Re > Re_{tr} \end{cases}, \quad (7.27a)$$

$$(7.27b)$$

where ρ is the density, u_n is the velocity component, and μ_t is the laminar (molecular) viscosity and all three of these quantities are for the local fluid conditions in the pump. The characteristic length (L) is input using array element PUMLEN(NP) for $NP=1,2,\dots,NPUMP$, and the constants d_t , d_l , and Re_{tr} are input using array elements ALPHAL(NC), ALPHAT(NC), and REYTRP(NC), respectively, for $NC=1,2,\dots,NCORRP$. The value input for array element ICORP(NP) sets the value of the index NC, as used above, for pump index number NP. At the present time, all pumps in a calculation must use the same set of characteristic correlations; therefore, the user must enter (a) ICORP(NP)=1 for $NP=1,2,\dots,NPUMP$, (b) NCORRP=1, and (c) values for ACORP(1), BCORP(1), CCORP(1), ALPHAL(1), ALPHAT(1), and REYTRP(1) which will be used for all pumps.

During a steady-state calculation, the relative pump flow rate ($\tilde{Q}$) used in Eq. 7.25 will be set internally by the code to 1.0; the relative pump speed ($\tilde{N}$) for pump index number NP will be set using Eq. 7.2 and the values input for the actual speed (N as input using array element PUMPS(NP)) and the reference speed (N_R as input using array element PUMPSR(NP)).

The final piece of user input required for a steady-state calculation is a value for IPUBUG, which controls the amount of print output generated by the subroutine which calculates pump quantities during the case. The nominal choice is IPUBUG=1; selecting IPUBUG=0 may result in the user being provided with too little information concerning the pumps.

Additional input, consisting of the final two variables shown in Table 7.5, is required for a transient calculation. This input should be provided when beginning a transient calculation from a steady-state calculation (i.e. ISTATE=2). (This information may be included in the input deck at the beginning of the steady-state calculation (i.e. ISTATE=0); then, if unchanged, the values do not need to be respecified at the beginning of the transient calculation.) The remainder of this section describes the additional input needed for a transient calculation.

During a transient calculation, the pump speed is a user-specified function of time (t) and is computed from

$$N = N_R * f_{nf}(t), \quad (7.28)$$

where $f_{nf}(t)$ is transient function number nf; the value corresponding to the subscript nf is input using array element NTPUMP(NP), for NP=1,2,...,NPUMP. If the speed for pump number NP is to be constant during a transient, the user must input a nonzero transient function number in array element NTPUMP(NP) and define this transient function as having a constant value (e.g. 1.0) for all time, rather than inputting NTPUMP(NP)=0. The speed from Eq. 7.28 is used, along with the reference speed as previously defined, in Eq. 7.2 to define the relative speed for use in computing the pressure increase from Eq. 7.25. During a transient calculation, the relative flow rate ($\tilde{Q}$) used in Eq. 7.25 is computed from Eq. 7.3, where Q is the current volumetric flow rate in the pump (computed internally by the code) and Q_R is the reference volumetric flow rate for pump NP which must be input using array element QFLOWR(NP).

7.4 Specified Pressure Increase Model

The specified pressure increase pump model is selected by IFPUMP=2 in NAMELIST DATA. The equations used in COMMIX to implement this model are presented in Sec. 8.4 of Ref. 1. The additional input to be provided by the user in NAMELIST PUMP is shown in Table 7.6. These input variables are discussed in the remainder of this section.

Unlike the homologous and specified speed models, the specified pressure model may only be used for a transient calculation. Additionally, the pumps must have existed in the steady-state calculation (and, thus, have been modeled using the homologous or specified speed models). The input variables shown in Table 7.6 may, therefore, only be specified when beginning a transient calculation from a steady-state calculation (i.e. ISTATE=2).

Table 7.6 Specified-Pressure-Increase Pump Model (IFPUMP=2)
Variables in NAMELIST PUMP¹

Variable	Meaning/Value	Restrictions
NTPUMP(NP)	nf: transient function number used to compute pump pressure increase.	$1 \leq NP \leq NPUMP$ $1 \leq NTPUMP \leq 25$
IPUBUG	select printing from subroutine PUMPQ (# of lines below is per pump, per outer iteration); may be: =0: none; ≥4: 1 line at exit.	

Notes:

1. This input block is present if, and only if, IFPUMP>0 and ISTATE=0 or 2; the choice of IFPUMP=2 is only allowed at ISTATE=2; the pump(s) must have been modeled during steady state using IFPUMP=1, 3, or 4.

As indicated by the name of this option, the user specifies the pressure increase to be provided by the pump (ΔP_p) as a function of time (t). This is implemented via the equation

$$\Delta P_p = \Delta P_{p,ss} * f_{nf}(t) , \quad (7.29)$$

where $\Delta P_{p,ss}$ is the pressure increase which was computed by the code for the pump during the steady-state calculation and f_{nf} is transient function number nf . The value entered in array element $NTPUMP(NP)$, for $NP=1,2,\dots, NPUMP$, in NAMELIST PUMP selects which transient function is to be used for pump number NP and is, thus, the value of the subscript nf in Eq. 7.29. If the pressure increase for pump number NP is to be constant during a transient, the user must input a nonzero transient function number in array element $NTPUMP(NP)$ and define this transient function as having a constant value (e.g. 1.0) for all time, rather than inputting $NTPUMP(NP)=0$.

The other piece of user input required is a value for $IPUBUG$ in NAMELIST PUMP, which controls the amount of print output generated by the subroutine which calculates pump quantities during the case. The nominal choice is $IPUBUG=4$; selecting $IPUBUG=0$ may result in the user being provided with too little information concerning the pumps.

7.5 Pump Location Input Block

Each pump, as described by the input in NAMELIST PUMP, must be associated with a particular momentum control volume in the computational grid. Identification of the momentum control volume involves two pieces of information: (1) the type of momentum control volume, as indicated by its orientation direction (i.e., either x (or r), y (or θ), or z) and (2) its spatial position in the grid. The type of momentum control volume also determines the pump orientation and pumping direction: flow through the pump is parallel to the direction of the momentum control volume. The pressure change provided by the pump model (i.e. Eq. 7.17, 7.25, or 7.29) is imposed in the direction of the pump orientation across the length of the momentum control volume in which the pump is located. This association of pumps with momentum control volumes is accomplished using the Pump Location input block.

The Pump Location input block is a set of fixed format records in the card-image input file. The structure of these records is shown in Table 7.7. The first field identifies the pump

orientation (i.e. the type of momentum control volume): a 'PUMX' record indicates the x (or r) direction, a 'PUMY' record indicates the y (or θ) direction, and a 'PUMZ' record indicates the z direction. The second field is the pump index number. The third, fifth, and seventh fields identify the pump location by its spatial indices (IB,JB,KB). (The values in the fourth, sixth, and eighth fields are, essentially, not used; however, values must be entered, and the values must be identical to the values entered in the third, fifth, and seventh fields, respectively.) The 'PUMX', 'PUMY', and 'PUMZ' records may be specified in any order. One, and only one, such record must be present in this input block for each pump index number ($1 \leq NP \leq NPUMP$) being used. A momentum control volume must not contain more than one pump. After locations have been specified for all pumps, an 'END ' record must be present to terminate this input block.

The geometry in the vicinity of the pump must be defined such that the flow through the pump is one dimensional and parallel to the direction specified by the pump orientation. This is accomplished by blocking the cell faces in the other two directions using (a) boundary surfaces (cf. Sec. 3.4) which have a velocity boundary condition (cf. Sec. 4.1.1) which gives zero flow across these faces and/or (b) surface porosities of zero (cf. Sec. 3.5). This process is most easily explained using an example, which is given in the next paragraph.

Consider a pump having an index number of 1, oriented in the z direction, and providing a pressure change between grid positions (I,J,K) of (3,4,5) and (3,4,6). The pump location is specified using the record

```
PUMZ  1  3  3  4  4  5  5
```

in the Pump Location input block. The channeling of the flow to be one dimensional in the region of the pump, using the latter of the two methods noted above, is accomplished using the records

```
ALX    0.0  2  3  4  4  5  6
ALY    0.0  3  3  3  4  5  6
```

Table 7.7 Pump Location Input Block Record Structure¹

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier, indicating the pump orientation; may be: 'PUMX': x (or r) direction; 'PUMY': y (or θ) direction; 'PUMZ': z direction; 'END ': terminates this input block; record positions 5-32 should be blank.	
5-8	I4	NUM	pump index number.	$1 \leq \text{NUM} \leq \text{NPUMP}$
9-12	I4	IB	I location index.	$1 \leq \text{IB} \leq \text{IMAX}$
13-16	I4	IE	I location index.	$\text{IE} = \text{IB}$
17-20	I4	JB	J location index.	$1 \leq \text{JB} \leq \text{IMAX}$
21-24	I4	JE	J location index.	$\text{JE} = \text{JB}$
25-28	I4	KB	K location index.	$1 \leq \text{KB} \leq \text{KMAX}$
29-32	I4	KE	K location index.	$\text{KE} = \text{KB}$
33-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. This input block is present if, and only if, $\text{IFPUMP} > 0$ and $\text{ISTATE} = 0$ or 2.
2. These records result in adding a pressure increase acting in direction "NAME" and described by pump number "NUM" to the "NAME"-direction momentum control volume at spatial position (IB,JB,KB) within the system volume.
3. The meaning of the indices (IB,JB,KB) depends on the orientation of the pump. For 'PUMX' records, these indices refer to the x (or r) momentum control volume which is located between fluid grid locations (IB,JB,KB) and (IB+1,JB,JB). For 'PUMY' records, these indices refer to the y (or θ) momentum control volume which is located between fluid grid locations (IB,JB,KB) and (IB,JB+1,KB). For 'PUMZ' records, these indices refer to the z momentum control volume which is located between fluid grid locations (IB,JB,KB) and (IB,JB,KB+1).
4. Use one 'PUMX', 'PUMY', or 'PUMZ' record to associate each pump with the proper momentum control volume. Each momentum control volume may contain, at most, one pump. The order of the records is irrelevant, except (a) if the same pump number is referenced by more than one record, the final such specification will be used, (b) if the same momentum control volume is reference by more than one record, the record having the lowest pump number will be used, and (c) the 'END ' record occurs only once and is last.
5. Pumps may only be placed in momentum control volumes which exist. For a momentum control volume to exist, the two main control volumes of which it is composed must both exist (i.e. contain fluid) and must not be separated from each other by a boundary surface. A momentum control volume which contains a pump should have a geometric description which only allows one-dimensional flow in the direction corresponding to the momentum control volume's direction.

in the Internal Cell Initialization input block. (Since surface porosities are input for main (rather than momentum) control volumes and a momentum control volume is composed of half of two

adjacent main control volumes, the surface porosities must be input as 0.0 for both axial levels K=5 and K=6.) When using one of the fluid turbulence models selected by ITURKE ≥ 10 , this total blocking of fluid cells must be accomplished using a pair of coincident boundary surfaces rather than specifying zero surface porosity (cf. Sec. 3.5). The user may want to specify additional surface and volume porosity information (to override the default of 1.0) in the region of the pump to further define the geometry. Recall (cf. Sec. 3.5) that the z-direction flow area (A_z for each cell face) and the fluid volume (V_f for each main control volume) are defined, assuming use of a Cartesian coordinate system, by

$$(A_z)_{i,j,k+1/2} = (\gamma_z)_{i,j,k+1/2} \Delta x_i \Delta y_j \quad (7.30)$$

and

$$(V_f)_{i,j,k} = (\gamma_v)_{i,j,k} \Delta x_i \Delta y_j \Delta z_k, \quad (7.31)$$

where γ_z is the z-direction surface porosity, γ_v is the volume porosity, and Δx , Δy , and Δz are the mesh spacings. To define a straight-duct pump region having an interior flow area of $0.25 \cdot \Delta x_3 \cdot \Delta y_4$ for the full height of axial levels K=5 and K=6, enter the records

ALZ	0.25	3	3	4	4	4	6
AL	0.25	3	3	4	4	5	6

in the Internal Cell Initialization input block. The flow area at the pump location is particularly important, since it is the product of this flow area and the fluid velocity which defines the volumetric flow rate (Q) used in the pump equations; for the present example, the volumetric flow rate is, thus, $A_z w$, where w is the fluid velocity component in the z direction.

8. TURBULENCE

8.1 Selection of Model

Modeling of fluid turbulence in COMMIX is accomplished by assuming the diffusion coefficients in the momentum and energy transport equations may be written as the sum of a laminar and a turbulent contribution (cf. Sec. 7 of Ref. 1). Specifically, the viscosity (μ) to be used as the diffusion coefficient in the momentum equation is assumed to be

$$\mu = \mu_l + \mu_t \quad (8.1)$$

and the thermal conductivity (λ) to be used in the diffusion coefficient ($\Gamma_h = \lambda/C_p$) for the energy equation is assumed to be

$$\lambda = \lambda_l + \lambda_t, \quad (8.2)$$

where the subscripts l and t stand for laminar (molecular) and turbulent properties, respectively. The laminar properties are provided directly by the fluid equation of state (cf. Sec. 9.1). The turbulent properties are a function of the flow field and are provided by one of several user-selectable models.

The value input for variable ITURKE in NAMELIST GEOM selects the manner in which the turbulent viscosity and the turbulent thermal conductivity values are provided. As shown in Table 8.1, there are five choices for ITURKE. The models selected by the various values of ITURKE and the additional input required for each model are discussed in the following subsections: the two "user-specified" models (ITURKE=0 and ITURKE=1) are in Sec. 8.2, the "k- ϵ " model (ITURKE=12) is in Sec. 8.3, the "k" model (ITURKE=11) is in Sec. 8.4, and the "mixing-length" model (ITURKE=10) is in Sec. 8.5. The case of "no turbulence modeling" is a special subcase of the "user-specified/constant-in-space" model (ITURKE=0) and is discussed in Sec. 8.2. The value of ITURKE must be specified at the beginning of a steady-state calculation (ISTATE=0); the value should not be changed at later stages in the calculational sequence (ISTATE>0).

Table 8.1 Turbulence Model Variables in NAMELIST GEOM

Variable	Meaning/Value	Restrictions
ITURKE	<p>selects type of model used for determining turbulent viscosity and conductivity for fluids; may be:</p> <p>= 0: user specified, constant in space;</p> <p>= 1: user specified, variable in space;</p> <p>= 10: 0-equation "mixing-length" model (not recommended);</p> <p>= 11: 1-equation "k" model (not recommended);</p> <p>= 12: 2-equation "k-ϵ" model</p>	ISTATE=0

There are several points the user should bear in mind when selecting a turbulence model. The accuracy of the turbulence modeling increases as the value of ITURKE increases. Increased accuracy is obtained at the expense of increased core storage and more computational work per outer iteration during the solution. As an example, the "k- ϵ " turbulence model (ITURKE=12) requires solving two systems of finite-difference equations (one for k and one for ϵ) which are in addition to those systems already being solved for the normal mass, momentum, and energy conservations. The user will normally want to chose a value of 0, 1, or 12 for ITURKE. There is little basis for chosing ITURKE=10 or ITURKE=11; these are simpler turbulence modeling approaches which were added to COMMIX prior to the implementation of the "k- ϵ " model.

Several prefacing remarks are appropriate before proceeding with the detailed input description. When using something other than the "user-specified" turbulence models (ITURKE $\geq$ 10), an impervious wall within the fluid domain must be represented as a pair of boundary surfaces (having unit surface normal vectors pointing in opposite directions and zero-slip velocity boundary conditions) rather than being represented merely by having a zero value for the directional surface porosity (cf. Secs 3 and 4). All of the turbulence models require one or more "constants" (e.g. characteristic length (D_h)) to be supplied by the user; at present, there is no provision for the user to specify these constants on a regional basis for the problem.

8.2 User-Specified Models

For the user-specified turbulence models, the user inputs the values for the turbulent viscosity and the turbulent thermal conductivity. These quantities are initially set using the input variables in NAMELIST DATA which are shown in Table 8.2. The turbulent viscosity value (μ_{t0}) is input using variable TURBV; in absence of experiment data, the turbulent viscosity may be estimated (Ref. 5, pp. 41-42) from

$$\mu_{t0} = 0.0028 C_{\mu 0} \rho u_{\max} D_h, \quad (8.3)$$

where

$$C_{\mu 0} = \begin{cases} 0 & \text{for } Re_{\max} < 1000 \\ 0.1 (0.001 Re_{\max} - 1) & \text{for } 1000 \leq Re_{\max} \leq 2000 \\ 0.1 & \text{for } 2000 < Re_{\max} \end{cases}, \quad (8.4)$$

Table 8.2 User-Specified Turbulence Model (ITURKE=0 or 1)
Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
TURBV	μ_{t0} : turbulent viscosity [Pa*s].	$TURBV \geq 0$
TURBC	λ_{t0} : turbulent thermal conductivity [W/m/°C].	$TURBC \geq 0$
CHARRE	Re_{ch} : characteristic Reynolds number [-] used in calculating turbulent thermal conductivity.	$TURBC=0$ $CHARRE > 0$
CHART	T_{ch} : characteristic temperature [°C] used in calculating turbulent thermal conductivity.	$TURBC=0$ $CHART \neq 0$

Notes:

1. If no fluid turbulence modeling is desired, input ITURKE=0 in NAMELIST GEOM and zeros for all four of the variables above.
2. When modeling fluid turbulence, input (1) $TURBV > 0$ and either (2a) $TURBC > 0$ with $CHARRE=0$ and $CHART=0$ or (2b) $TURBC=0$ with $CHARRE > 0$ and $CHART \neq 0$ (cf. Eq. 8.5).

ρ is the fluid density, D_h is a characteristic length, $u_{\max}$ is the maximum fluid velocity, and $Re_{\max}$ is the maximum fluid Reynolds number. The turbulent thermal conductivity (λ_{t0}) is set in one of two ways depending on the value input for variable TURBC. If $TURBC > 0$ is input, then the value input is used directly as the turbulent thermal conductivity; in this case, values should not be input for variables CHARRE and CHART. If $TURBC = 0.0$ is input, then the turbulent thermal conductivity is computed by the code from

$$\lambda_{t0} = \frac{C_p \mu_{t0}}{0.8 / [1 - \exp(-6 \cdot 10^{-5} Re_{ch} Pr_{ch}^{0.333})]}, \quad (8.5)$$

where the characteristic Reynolds number (Re_{ch}) is input using variable CHARRE and the specific heat capacity (C_p) and the characteristic Prandtl number ($Pr_{ch} = C_p \mu_t / \lambda_t$) are computed by the code using the equation-of-state information (cf. Sec. 9.1) for fluid number 1 evaluated at the characteristic temperature (T_{ch}) input using variable CHART.

If the turbulent viscosity and the turbulent thermal conductivity values are to be constant spatially (selected by $ITURKE = 0$), then the above paragraph describes all the input required. The values of $TURBV$ and $TURBC$ (or $CHARRE$ and $CHART$) must be entered at the beginning of a steady-state calculation ($ISTATE = 0$); if necessary, revised values of $TURBV$ and $TURBC$ may input during continuation runs ($ISTATE > 0$). The values of μ_{t0} and λ_{t0} are used for μ_t in Eq. 8.1 and λ_t in Eq. 8.2, respectively, for all fluid cells.

The case of no turbulence modeling is a special subcase of the "user-specified/constant-in-space" model for which the turbulent viscosity and the turbulent thermal conductivity are both zero. In this case, Eqs. 8.1 and 8.2 reduce to $\mu = \mu_t$ and $\lambda = \lambda_t$, respectively. The user inputs $ITURKE = 0$ in NAMELIST GEOM and $TURBV = 0.0$ and $TURBC = 0.0$ in NAMELIST DATA.

Provision has been made for the user to specify a spatial distribution of turbulent viscosity and turbulent thermal conductivity values. This option is selected by inputting $ITURKE = 1$ in NAMELIST GEOM. The user inputs values for $TURBV$ and $TURBC$ (or $CHARRE$ and $CHART$) in NAMELIST DATA as described above. The resultant turbulent viscosity and

turbulent thermal conductivity values are used to initialize default values for all fluid cells at the beginning of a steady-state calculation (ISTATE=0). The user may then override these default values on a cell-by-cell basis by including the records in the Internal Cell Initialization Input Block which are shown in Table 8.3. (Recall that this input block is used to provide initial values for numerous variables, as discussed in Sec. 4.2.2.) The user may input as many 'TURV' and 'TURC' records as needed to provide values for specific fluid cells; these values should be input at the beginning of a steady-state calculation (ISTATE=0). If necessary, the user may override these values during a continuation run (ISTATE>0); these changes may only be accomplished by using the 'TURV' and 'TURC' records in the Internal Cell Initialization Input Block -- attempts to enter new values using variables TURBV and TURBC in NAMELIST DATA will be ignored at ISTATE>0 when ITURKE=1.

8.3 Two-Equation k-ε Model

For the k-ε turbulence model, the code solves two additional systems of transport equations: one for the turbulent kinetic energy (k) and one for the turbulent kinetic energy dissipation rate (epsilon). These equation systems are described in Sec. 7.3 of Ref. 1. After these equation systems are solved (for every fluid cell at every time step), the results are used to compute the turbulent viscosity and the turbulent thermal conductivity from the following equations:

$$\mu_t = \frac{f_\mu C_\mu \rho k^2}{\epsilon} \quad (8.6a)$$

and

$$\lambda_t = \frac{C_p \mu_t}{\sigma_h}, \quad (8.7)$$

where the turbulent viscosity coefficient function is defined by

$$f_\mu = \left[1 - \exp \left(- \frac{\rho k^2}{26.5 \mu_t \epsilon} \right) \right]^2. \quad (8.8)$$

**Table 8.3 User-Specified Turbulence Model (ITURKE=1) Variables
in Internal Cell Initialization Input Block**

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier; may be: 'TURC': "VALUE" is turbulent thermal conductivity [W/m/°-C]; 'TURV': "VALUE" is turbulent viscosity [Pa*s]; 'END ': terminates this input block; record positions 5-38 should be blank.	
5-14	F10.4	VALUE	see NAME.	
15-18	I4	IB	beginning I-range index.	$1 \leq IB \leq IE$
19-22	I4	IE	ending I-range index.	$IB \leq IE \leq IMAX$
23-26	I4	JB	beginning J-range index.	$1 \leq JB \leq JE$
27-30	I4	JE	ending J-range index.	$JB \leq JE \leq JMAX$
31-34	I4	KB	beginning K-range index.	$1 \leq KB \leq KE$
35-38	I4	KE	ending K-range index.	$KB \leq KE \leq KMAX$
39-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. These records result in assigning the value "VALUE" to the variable named "NAME" for the spatial positions (I,J,K) in the three-dimensional region delimited by $IB \leq I \leq IE$, $JB \leq J \leq JE$, and $KB \leq K \leq KE$ within the system volume.
2. Use as many 'TURC' and 'TURV' records as needed to provide pointwise initializations of the indicated variables which override the automatic initializations performed by the code; the order of the records is irrelevant, except if a cell is assigned a value in more than one record, the final such specification will be used.
3. The indices (I,J,K) refer to cell-center grid points; the cells must actually exist (i.e., contain fluid).
4. These 'TURC' and 'TURV' records may be interspersed among the other records (i.e. 'AL ', 'ALX ', 'ALY ', 'ALZ ', 'P ', 'QSOU', 'TL ', 'UL ', 'VL ', 'WL ') in the Internal Cell Initialization Input Block (cf. Tables 3.10 and Table 4.10). An 'END ' record terminates the input block, occurs only once, and must be last.

The k- ϵ turbulence model is selected by inputting ITURKE=12 in NAMELIST GEOM. The additional user input required by this model is provided using the variables in NAMELIST DATA which are shown in Table 8.4; the terminology used in this table corresponds to that used in the model development in Ref. 1. Six of these variables provide "modeling constants": CDTURB is C_μ , CT1 is C_1 , CT2 is C_2 , PRNDLD is σ_ϵ , PRNDLH is σ_h , and PRNDLK is σ_k ; these constants are generally assumed to be problem independent; default values are provided by the code and the values are shown in the Input Guide (Appendix A).

Two of the variables in Table 8.4 are specifically problem dependent: TDIN and TKIN. The initial (k_0) and inlet (k_{in}) values of the turbulent kinetic energy are computed by the code from

$$k_0 = C_{k,in} (\bar{u}_0^2 + \bar{v}_0^2 + \bar{w}_0^2) \quad (8.9)$$

and

$$k_{in} = C_{k,in} u_n^2, \quad (8.10)$$

respectively, where $\bar{u}_0$, $\bar{v}_0$, and $\bar{w}_0$ are the cell-center averages of the initial values of the three velocity components, u_n is the normal velocity at an inlet boundary surface (e.g. having KFLOW=1 or 100+n), and the value of $C_{k,in}$ is input using variable TKIN. The initial (ϵ_0) and inlet (ϵ_{in}) values of the turbulent kinetic energy dissipation rate are computed by the code from

$$\epsilon_0 = C_{\epsilon,in} k_0^{1.5} \quad (8.11)$$

and

$$\epsilon_{in} = C_{\epsilon,in} k_{in}^{1.5} \quad (8.12)$$

respectively, where the value of $C_{\epsilon,in}$ is input using variable TDIN. (Note that $C_{\epsilon,in}$ has units of inverse length.) Equations 8.9 and 8.11 are only invoked at the beginning of a steady-state calculation (ISTATE=0); during a continuation run (ISTATE>0), the initial values of the turbulent kinetic energy and the turbulent kinetic energy dissipation rate are read from the restart

Table 8.4 Two-Equation ("k-ε") Turbulence Model (ITURKE=12) Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
CDTURB	C_μ : constant in turbulence equations [-].	CDTURB > 0
CT1	C_1 : constant in source term in turbulent kinetic energy dissipation rate transport equation [-].	CT1 > 0
CT2	C_2 : constant in source term in turbulent kinetic energy dissipation rate transport equation [-].	CT2 > 0
PRNDLD	σ_ϵ : effective turbulent Prandtl number for turbulent kinetic energy dissipation rate [-].	PRNDLD > 0
PRNDLH	σ_h : effective turbulent Prandtl number for energy [-].	PRNDLH > 0
PRNDLK	σ_k : effective turbulent Prandtl number for turbulent kinetic energy [-].	PRNDLK > 0
TDIN	$C_{\epsilon, \text{in}}$: constant [m^{-1}] in equation which sets initial and inlet values of the turbulent kinetic energy dissipation rate: $\epsilon = C_{\epsilon, \text{in}} * k^{1.5}$.	TDIN > 0
TKIN	$C_{k, \text{in}}$: constant [-] in equations which set initial (i.v.) and inlet (b.v.) values of the turbulent kinetic energy: i.v.: $k = C_{k, \text{in}} * (u_0^2 + v_0^2 + w_0^2)$ b.v.: $k = C_{k, \text{in}} * (u_n^2)$.	TKIN > 0
EPS6	ϵ_k and ϵ_ϵ : inner iteration convergence criterion for turbulent kinetic energy and turbulent kinetic energy dissipation rate [-].	0 < EPS6 < 1
ITMAXK	maximum number of inner iterations (per outer iteration) allowed to solve the turbulent kinetic energy and turbulent kinetic energy dissipation rate equation systems.	ITMAXK > 0
NTJUMP	frequency with which the turbulence equations are solved, i.e. every NTJUMP-th time step.	NTJUMP > 0
RELAXK	ω_k and ω_ϵ : relaxation factor for solution of the turbulent kinetic energy and the turbulent kinetic energy dissipation rate transport equations [-].	0 < RELAXK ≤ 1
OMEGAD	Ω_ϵ : relaxation factor for coefficients in the turbulent kinetic energy dissipation rate transport equation [-].	0 < OMEGAD ≤ 1
OMEGAK	Ω_k : relaxation factor for coefficients in the turbulent kinetic energy transport equation [-].	0 < OMEGAK ≤ 1
OMEGAT	Ω_v : underrelaxation factor for advancement of turbulent viscosity across a time step [-].	0 < OMEGAT ≤ 1

file. When the inlet velocity is uniform and specific information for the turbulence quantities is not known, a recommendation (Ref. 5, p. 46) is that

$$C_{k,in} = 0.001 \quad (8.13)$$

and

$$C_{\epsilon,in} = \frac{C_{\mu}}{0.04025 D_h} , \quad (8.14)$$

where D_h is a characterisitic length, to be used as estimates.

The remaining seven variables in Table 8.4 are associated with the numerics of solving the turbulence transport equations. The two systems of finite-difference equations are solved in separate blocks of the calculational sequence (cf. Table 11.2 of Ref. 1) using the successive-overrelaxation numerical method (cf. Sec. 11.5 of Ref. 1). The relaxation factors input using variables OMEGAK, OMEGAD, and RELAXK are analogous to those input (OMEGAE and RELAXE) for use in solving the fluid energy equation system and are discussed in Sec. 12.4. Inner iterations continue until (a) the maximum relative change between successive inner iterations is less than the value of input variable EPS6 (ϵ_{ϕ} in Eq. 11.18 of Ref. 1, where $\phi=k$ or ϵ) or (b) the maximum number of inner iterations specified using input variable ITMAXK is reached. Clearly, there is a relationship between EPS6 and ITMAXK: as the value of EPS6 decreases, the value of ITMAXK must be increased in order for the desired degree of convergence to actually be achieved. Input variable NTJUMP specifies the frequency with which these equation systems are solved; the equations should normally be solved at every time step (NTJUMP=1). Input variable OMEGAT is an underrelaxation factor (Ω_t) used in advancing the turbulent viscosity values across a time step. If $0 < \Omega_t < 1$, then the turbulent viscosity is computed from

$$\mu_t = (1 - \Omega_t) \mu_t^{n-1} + \Omega_t \frac{f_{\mu} C_{\mu} \rho k^2}{\epsilon} , \quad (8.6b)$$

where the superscript n-1 in the first term denotes values from the end of the previous time step

(i.e. at t^{n-1}) and the absence of superscripts in the second term denotes the values are for the end of the current time step (i.e. at t^n); if $\Omega_i=1$, Equation 8.6b reduces to Eq. 8.6a. (This approach is the same as that used to advance the radiation heat flux (cf. Eq. 6.48 of Ref. 1).) Ideally, the underrelaxation factor is one; in practice, the underrelaxation factor may need to be less than one, particularly during the early stages of a steady-state calculation when the turbulence quantities are undergoing significant changes from their guessed initial values. (Additional discussion of iteration-control parameters is given in Sec. 12.4.)

All of the input variables in Table 8.4 should be specified at the beginning of a steady-state calculation (ISTATE=0). If necessary, revised values may be input during a continuation run (ISTATE>0).

8.4 One-Equation k Model

For this turbulence model, the code solves one additional system of transport equations for the turbulent kinetic energy (k). After this equation system has been solved (for every fluid cell at every time step), the turbulent kinetic energy dissipation rate (ε) is computed from the algebraic expression

$$\varepsilon = \frac{C_\mu^{0.75} k^{1.5}}{\kappa \ell} \quad (8.15)$$

for each fluid cell (rather than solving a second system of transport equations as is done for the k - ε turbulence model). The length scale (ℓ) in this equation is selected by the code to be the smaller of (a) the distance from the mesh point (center of the fluid cell) to the nearest no-slip rigid-wall boundary and (b) $0.175 \cdot D_h$. The value of the von Karman constant (κ) is set internally by the code to $0.23/C_\mu^{0.25}$ (and has a value of 0.42 when C_μ has the default value of 0.09). The characteristic length (D_h) and the constant C_μ are input by the user. After k and ε are computed, the turbulent viscosity (μ_t) and the turbulent thermal conductivity (λ_t) are computed from Eqs. 8.6 and 8.7, respectively.

This turbulence model is selected by inputting ITURKE=11 in NAMELIST GEOM. The

additional user input required by this model is provided using the variables in NAMELIST DATA which are shown in Table 8.5. Input variable HYDIN is the characteristic length (D_h) used in connection with computing the length scale (ℓ) (cf. note (b) following Eq. 8.15). The other input variables in Table 8.5 are merely a subset of the input required for the k- ϵ turbulence model, and the user is referred to Sec. 8.3 for their meanings.

Table 8.5 One-Equation ("k") Turbulence Model (ITURKE=11) Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
CDTURB	C_μ : constant in turbulence equations [-].	CDTURB > 0
PRNDLH	σ_h : effective turbulent Prandtl number for energy [-].	PRNDLH > 0
PRNDLK	σ_k : effective turbulent Prandtl number for turbulent kinetic energy [-].	PRNDLK > 0
HYDIN	D_h : characteristic length (or hydraulic diameter) [m]. (If IGEOM > 0, HDIN will be computed by the code.)	HYDIN > 0
TDIN	$C_{\epsilon, \text{in}}$: constant [m^{-1}] in equation which sets initial and inlet values of the turbulent kinetic energy dissipation rate: $\epsilon = C_{\epsilon, \text{in}} * k^{1.5}$.	TDIN > 0
TKIN	$C_{k, \text{in}}$: constant [-] in equations which set initial (i.v.) and inlet (b.v.) values of the turbulent kinetic energy: i.v.: $k = C_{k, \text{in}} * (u_0^2 + v_0^2 + w_0^2)$ b.v.: $k = C_{k, \text{in}} * (u_n^2)$.	TKIN > 0
EPS6	ϵ_1 : inner iteration convergence criterion for turbulent kinetic energy [-].	$0 < \text{EPS6} < 1$
ITMAXK	maximum number of inner iterations (per outer iteration) allowed to solve the turbulent kinetic energy equation system.	ITMAXK > 0
NTJUMP	frequency with which the turbulence equations are solved, i.e. every NTJUMP-th time step.	NTJUMP > 0
RELAXK	ω_k : relaxation factor for solution of the turbulent kinetic energy transport equation [-].	$0 < \text{RELAXK} \leq 1$
OMEGAK	Ω_k : relaxation factor for coefficients in the turbulent kinetic energy transport equation [-].	$0 < \text{OMEGAK} \leq 1$
OMEGAT	Ω_v : underrelaxation factor for advancement of turbulent viscosity across a time step [-].	$0 < \text{OMEGAT} \leq 1$

8.5 Zero-Equation Mixing-Length Model

For this turbulence model, the code does not solve the additional systems of transport equations for the turbulent kinetic energy or the turbulent kinetic energy dissipation rate. Instead, the turbulent viscosity is computed from the local velocity gradients using the algebraic expression

$$\mu_t = \rho(\kappa\ell)^2 \left[\frac{\partial u_i}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right]^{0.5} \quad (8.16)$$

for each fluid cell, where u_i is the velocity component in spatial direction x_i and the repeated appearance of a subscript (i and j) indicates a summation over all three spatial directions (e.g. x , y , and z). (The expanded form of the summation term is given in Table 7.1 of Ref. 1 for Cartesian and cylindrical coordinates.) The von Karman constant (κ) and the length scale (ℓ) are the same as defined in Sec. 8.4. The turbulent thermal conductivity is computed from Eq. 8.7 for each fluid cell.

This turbulence model is selected by inputting ITURKE=10 in NAMELIST GEOM. The additional user input required by this model is provided using the variables in NAMELIST DATA which are shown in Table 8.6. Input variable HYDIN is the characteristic length (D_h) used in connection with computing the length scale (ℓ) (cf. note (b) following Eq. 8.15); the other two input variables in Table 8.6 have the same meanings as in Sec. 8.3.

Table 8.6 Zero-Equation "Mixing-Length" Turbulence Model (ITURKE=10) Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
PRNDLH	σ_h : effective turbulent Prandtl number for energy [-].	PRNDLH > 0
HYDIN	D_h : characteristic length (or hydraulic diameter) [m]. (If IGEOM > 0, HYDIN will be computed by the code.)	HYDIN > 0
OMEGAT	Ω_t : underrelaxation factor for advancement of turbulent viscosity across a time step [-].	0 < OMEGAT ≤ 1

9. PROPERTIES AND HEAT TRANSFER COEFFICIENTS

The user must provide COMMIX with the physical properties of all fluids, the physical properties of all structure materials, and all fluid/structure heat transfer coefficients used in a calculation. The user inputs required for each of these categories are described in the following subsections.

9.1 Fluid Properties

The fluid transport equations contain numerous state and transport properties. Only two of the state properties may be considered independent; in COMMIX these are the enthalpy, h , and the pressure, P . An equation of state must be provided in order to calculate other dependent state properties (e.g. density, ρ , and temperature, T) from the independent state properties. The values of the laminar (molecular) transport properties (e.g. viscosity, μ_l , and thermal conductivity, λ_l) must also be provided. (As discussed in Sec. 8, the values of the turbulent transport properties are either provided by the user or calculated by the code.) The current formulation also requires certain derivatives of properties (e.g. derivative of enthalpy with respect to temperature, C_p , and derivative of density with respect to enthalpy) and the saturation pressure, P_{sat} . (The saturation pressure is only used within the main part of COMMIX in connection with an option which checks to see whether saturation conditions have been achieved in fluid number 1, which is discussed in Sec. 12.1.)

This fluid property information must be provided by the user for each fluid number m ($1 \leq m \leq \text{IFHTX}$) used in a calculation. (The use of multiple fluids and the input of the number of fluids being used (IFHTX) is discussed in Sec. 3.3.) The user provides this information in two ways: source (or object) code and the values of input variables in NAMELIST DATA which are shown in Table 9.1.

There are two methods for the user to describe the properties of fluid number 1. The choice between these two methods is indicated by the value input for variable IFPROP in NAMELIST DATA: IFPROP=0 selects the "detailed" property method and IFPROP=1 selects the

Table 9.1 Fluid Property Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
IFPROP	selects which type of properties to use for fluid 1; may be: 0: detailed; supply source/object code as directed in text; 1: simplified; supply values for element 1 of the next 9 arrays.	
<p>Simplified properties for fluid m are assumed to have the following functional forms:</p> $h = C_{0hm} + C_{1hm} * T,$ $\rho = C_{0\rho m} + C_{1\rho m} * T \text{ or}$ $\rho = P / [(8314 / C_{2\rho m}) * (T + 273.15)],$ $\lambda_t = C_{0\lambda m} + C_{1\lambda m} * T, \text{ and}$ $\mu_t = C_{0\mu m} + C_{1\mu m} * T$ <p>where h is enthalpy [J/kg], ρ is density [kg/m³], λ_t is laminar (molecular) thermal conductivity [W/m · °C], μ_t is laminar (molecular) viscosity [Pa · s], T is temperature [°C], and P is pressure [Pa]. The choice between density functions is made based on the value input for $C_{2\rho m}$. The next 9 variables must be input (a) for fluid 1 if it uses simplified properties and (b) for all other fluid numbers m.</p>		m=1: IFPROP=1 2 ≤ m ≤ IFHTX
FC0H(m)	C_{0hm} : additive constant [J/kg] in enthalpy function for fluid m.	
FC1H(m)	C_{1hm} : temperature multiplier (i.e. specific heat capacity) [J/kg · °C] in enthalpy function for fluid m.	FC1H > 0
FC0RO(m)	$C_{0\rho m}$: additive constant [kg/m ³] in first density function for fluid m.	See note 1.
FC1RO(m)	$C_{1\rho m}$: temperature multiplier [kg/m ³ · °C] in first density function for fluid m.	See note 1.
GASMW(m)	$C_{2\rho m}$: molecular weight [kg/kg · mole] in second density function (i.e. ideal gas) for fluid m.	See note 1.
FC0K(m)	$C_{0\lambda m}$: additive constant [W/m · °C] in thermal conductivity function for fluid m.	
FC1K(m)	$C_{1\lambda m}$: temperature multiplier [W/m · °C ²] in thermal conductivity function for fluid m.	
FC0MU(m)	$C_{0\mu m}$: additive constant [Pa · s] in viscosity function for fluid m.	
FC1MU(m)	$C_{1\mu m}$: temperature multiplier [Pa · s/°C] in thermal conductivity function for fluid m.	
FCTLO	T_{low} : lower temperature limit [°C] for property table printing.	See note 2.
FCTHI	T_{high} : upper temperature limit [°C] for property table printing.	See note 2.

Notes:

1. For each fluid m, input either (a) meaningful values of FC0RO(m) and FC1RO(m) with GASMW(m)=0 or (b) meaningful value of GASMW(m) (>0) with FC0RO(m)=0 and FC1RO(m)=0.
2. Only affects printing of tables showing property values at pressure PRES0(m) for each fluid number m for the temperature range between FCTLO and FCTHI in 10 equal-increment steps; FCTLO < FCTHI.

"simplified" property method. All other fluids (i.e. fluid number 2 through fluid number IFHTX) used in a calculation must be described using the "simplified" property method. The user input required for the detailed and simplified property methods are described in the following three paragraphs, followed by a discussion of the advantages and disadvantages of each method.

The "detailed" fluid properties are specified using a set of nine FORTRAN FUNCTIONS. These routines must conform to the functional dependencies, FUNCTION names, and argument orderings shown in Table 9.2. The first eight of these FUNCTIONS contain fits to the available physical property data. The terminology "h or T" in the functional dependency means that the correlation may be based on either enthalpy or temperature; both of these quantities are passed as arguments to the corresponding FUNCTION. The eighth function in Table 9.2 must be the exact inverse of the second function; i.e.

$$h = \text{HLIQ}(P, \text{TLIQ}(h, P, T_{\text{guess}})) \quad (9.1)$$

for all values of h and P. The ninth function in Table 9.2 provides an eight-character description of the fluid (FLUDID) and establishes the default temperature limits (T_{low} and T_{high}); these three quantities are only used in connection with printing a small table showing property values at the beginning of a COMMIX run; the values of T_{low} and T_{high} are not used to indicate the range over which the fluid properties are valid.

The way in which the user supplies to COMMIX the FUNCTIONS describing "detailed" fluid properties depends on the way in which COMMIX has been installed on the user's computer system. The details of these installation and execution procedures are deferred to Appendix C; however, the user must ultimately take some action to insure that the object code for the FUNCTIONS describing the detailed properties for the desired fluid is linked with the object code for the rest of COMMIX when forming a complete executable code version. Only one set of these detailed fluid property FUNCTIONS may be linked with the code at a given time, and these detailed properties may only be used for fluid number 1.

The "simplified" fluid properties are computed using a set of STATEMENT FUNCTIONS provided in the source code having the following general forms:

$$h = C_{0hm} + C_{1hm}T, \quad (9.2)$$

$$\rho = C_{0\rho m} + C_{1\rho m}T \quad (9.3a)$$

or

$$\rho = \frac{P}{\left(\frac{8314}{C_{2\rho m}}\right)(T + 273.15)} \quad (9.3b)$$

$$\lambda_t = C_{0\lambda m} + C_{1\lambda m}T, \quad (9.4)$$

and

$$\mu_t = C_{0\mu m} + C_{1\mu m}T, \quad (9.5)$$

Table 9.2 Functional Dependence for the Detailed Fluid Properties and the Names and Arguments of the Corresponding FORTRAN FUNCTIONS

Theory		FORTRAN FUNCTION	
Property [Units]	Dependence	Name	Arguments
ρ [kg/m ³]	(h or T,P)	ROLIQ	(T,h,P)
T [°C]	(h,P)	TLIQ	(h,P,T _{guess})
μ_t [Pa·s]	(h or T,P)	VISLIQ	(T,h,P)
λ_t [W/m·°C]	(h or T)	THCLIQ	(T,h)
C_p [J/kg·°C]	(h or T,P)	CPI LIQ	(T,h,P)
$d\rho/dh$ [kg ² /m ³ ·J]	(h or T,P)	DRODHL	(T,h,P)
P_{sat} [Pa]	(T)	PSAT1	(T)
h [J/kg]	(T,P)	HLIQ	(P,T)
fluid identification [-]	--	FLUDID	(T _{low} , T _{high})

Notes:

1. All FUNCTIONS and their arguments are type REAL (which means DOUBLE PRECISION on "short-word" machines), except FLUDID is CHARACTER*8.
2. Units for function arguments are temperature T in [°C], enthalpy h in [J/kg], and pressure P in [Pa].
3. All arguments are input to the FUNCTIONS, except T_{low} and T_{high} are returned by FLUDID.
4. The argument T_{guess} in TLIQ is an estimate of the temperature and is used (if necessary) to start an iterative process to numerically invert HLIQ.

where the properties have the normal COMMIX units (i.e. temperature T in $^{\circ}\text{C}$, pressure P in $[\text{Pa}]$, etc.). The two constants appearing in the ideal-gas-law form of the density function (Eq. 9.3b) are the universal gas constant ($R_0=8314 \text{ J/kg-mole}\cdot\text{K}$) and the units conversion (273.15) from $^{\circ}\text{C}$ to $[\text{K}]$. The user inputs values for the coefficients C_{ipm} using variables $\text{FC0H}(m)$, $\text{FC1H}(m)$, $\text{FC0RO}(m)$, $\text{FC1RO}(m)$, $\text{GASMW}(m)$, $\text{FC0K}(m)$, $\text{FC1K}(m)$, $\text{FC0MU}(m)$, and $\text{FC1MU}(m)$ in NAMELIST DATA (cf. Table 9.1) for each fluid number m which is to use simplified properties. The choice between the two forms of the density function is made based on the value input for $C_{2\rho m}$, which corresponds to the fluid's molecular weight: if $C_{2\rho m}=0$, the linear form given by Eq. 9.3a is used; otherwise, the ideal-gas-law form given by Eq. 9.3b is used. The forms given by Eqs. 9.2 through 9.5 provide four of the needed fluid property functions; the others are computed as follows: (a) the constants input for use in the $h(T)$ function (Eq. 9.2) are used internally by the code to generate the inverse function $T(h)$; (b) the specific heat capacity C_p is given by the input constant C_{1hm} ; (c) the derivative $d\rho/dh$ is computed by dividing $d\rho/dT$ (from differentiation of Eq. 9.3) by dh/dT (from differentiation of Eq. 9.2). The saturation pressure and fluid identification functions are not used for simplified properties.

The user needs to consider several things in deciding whether to use detailed or simplified properties for fluid number 1: accuracy, execution speed, and ease of use. The detailed properties method has the most flexibility, allowing the user to achieve a high degree of accuracy in representing the physical property variations using numerical fits; the simplified properties method is restricted to linear forms (or an ideal-gas-law form for the density). Use of simplified properties results in somewhat faster code execution than use of detailed properties, since the coding for the STATEMENT FUNCTIONS is expanded "in line" during compilation rather than the issuing of a "return-jump" to a FUNCTION entry point. Finally, the user may choose simplified properties because detailed property FUNCTIONS have not been coded for a particular fluid. With regard to this last item, four sets of FUNCTIONS describing detailed properties for liquid sodium, liquid normal water, liquid heavy water, and helium gas have been coded and are distributed as part of the COMMIX source code package; the specific functional forms used for each of these four fluids are presented in Appendix B; the user may use the source code for these FUNCTION sets as a guide in creating detailed property packages for other fluids.

At the beginning of each run, COMMIX will print a short table summarizing the property values for each fluid. Input variables FCTLO and FCTHI in NAMELIST DATA (cf. Table 9.1) are the temperature limits used in printing this table, and, if input, override the defaults (T_{low} and T_{high} , respectively) set by FUNCTION FLUIDID.

9.2 Structure Properties

The thermal effects of structures surrounding, or imbedded in, the fluid are included using part of the fluid energy boundary conditions and using the thermal structure model; the majority of the input associated with these models was described in Sec. 4.1.3 and Sec. 5, respectively. In order for these effects to be computed, the code must have values of the density (ρ), specific heat capacity (C_p), and thermal conductivity (λ) for each structure. The user provides this information in two ways: the values of the input variables in NAMELIST DATA shown in Table 9.3 and the optional provision of source (or object) code.

Each structure material is identified by a structure material index number; the structure material associated with boundary surface ns is identified by the value specified in array element MATWAL(ns) in NAMELIST DATA (cf. Table 4.6); the structure material composing a material region of a thermal structure is identified by the value specified for variable MI in NAMELIST M in the Thermal Structure Prototype input block (cf. Table 5.4). The same structure material index number may be used in more than one physical structure. The user inputs in variable NMATER in NAMELIST DATA (cf. Table 9.3) the maximum of the values entered in MATWAL(ns) and MI. (Although not a requirement, the user will find it most convenient to use structure material index numbers in a contiguous sequence beginning with 1; in this case, NMATER will be the total number of structure materials.) The maximum value allowed for NMATER is 20.

The physical properties of solid materials are nominally assumed to have the following forms:

$$\rho = C_{0\rho m} + C_{1\rho m}T + C_{2\rho m}T^2, \quad (9.6)$$

Table 9.3 Structure (Solid) Property Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
NMATER	number of structure materials (i.e. maximum of values entered for MATWAL(N) in NAMELIST DATA and MI in NAMELISTs M of Thermal Structure Prototype input block).	$0 < \text{NMATER} \leq 20$
<p>Properties for structure material number m are assumed to have the following functional forms:</p> $\rho = C_{0\rho m} + C_{1\rho m} * T + C_{2\rho m} * T^2,$ $C_p = C_{0Cm} + C_{1Cm} * T + C_{2Cm} * T^2, \text{ and}$ $\lambda = C_{0\lambda m} + C_{1\lambda m} * T + C_{2\lambda m} * T^2$ <p>where ρ is density [kg/m³], C_p is specific heat capacity [J/kg · °C], λ is thermal conductivity [W/m · °C], and T is temperature [°C]. The next 9 variables must be input for each structure material number m.</p>		$1 \leq m \leq \text{MATER}$
C0RO(m)	$C_{0\rho m}$: additive constant [kg/m ³] in density function for material m.	
C1RO(m)	$C_{1\rho m}$: multiplier [kg/m ³ · °C] of T in density function for material m.	
C2RO(m)	$C_{2\rho m}$: multiplier [kg/m ³ · °C ²] of T ² in density function for material m.	
C0CP(m)	C_{0Cm} : additive constant [J/kg · °C] in specific heat capacity function for material m.	
C1CP(m)	C_{1Cm} : multiplier [J/kg · °C ²] of T in specific heat capacity function for material m.	
C2CP(m)	C_{2Cm} : multiplier [J/kg · °C ³] of T ² in specific heat capacity function for material m.	
C0K(m)	$C_{0\lambda m}$: additive constant [W/m · °C] in thermal conductivity function for material m.	
C1K(m)	$C_{1\lambda m}$: multiplier [W/m · °C ²] of T in thermal conductivity function for material m.	
C2K(m)	$C_{2\lambda m}$: multiplier [W/m · °C ³] of T ² in thermal conductivity function for material m.	

Notes:

1. The user may provide alternate forms for C_p and λ by supplying alternate coding for FUNCTIONS CPSOL1 and XKSOL1, respectively.

$$C_p = C_{0Cm} + C_{1Cm}T + C_{2Cm}T^2, \quad (9.7)$$

and

$$\lambda = C_{0\lambda m} + C_{1\lambda m}T + C_{2\lambda m}T^2, \quad (9.8)$$

where the properties have the normal COMMIX units (i.e. temperature T in [$^{\circ}\text{C}$], density ρ in [kg/m^3], etc.). The user inputs values for the coefficients C_{ipm} using variables C0RO(m), C1RO(m), C2RO(m), C0CP(m), C1CP(m), C2CP(m), C0K(m), C1K(m), and C2K(m) in NAMELIST DATA (cf. Table 9.3) for each structure material number m which is used in the calculation.

The specific heat capacity and thermal conductivity are actually computed by FORTRAN FUNCTIONS CPSOL1 and XKSOL1, respectively, whose entry points are described in Tables 9.4 and 9.5, respectively. With this entry-point knowledge, the user may write modified versions of these two FUNCTIONS (and compile and link them to the rest of COMMIX (cf. Appendix C.2)) in order to specify different functional forms for materials whose properties are not well described by the basic quadratic form. (The user is not expected to need this capability to code modified forms very often.)

The user is cautioned on one aspect of specifying the structure material density. The volume of each element of structure is constant during a calculation. A temperature-dependent density means that the mass of the structure element changes as the temperature changes. Since this is probably not a desired behavior, the user will typically want to use a constant density (i.e. input C1RO(m)=0 and C2RO(m)=0) for each structure material (especially for a transient). (With this consideration in mind, the user has not been provided with a capability (similar to that afforded by FUNCTIONS CPSOL1 and XKSOL1) to override the quadratic form given by Eq. 9.6 for the structure material density.)

Table 9.4 Entry Point Description for FUNCTION CPSOL1

Entry Point:
 FUNCTION CPSOL1(IRM,T,C0CP,C1CP,C2CP)
 INTEGER IRM
 REAL CPSOL1,T,C0CP,C1CP,C2CP
 DIMENSION C0CP(20),C1CP(20),C2CP(20)

Code	Text	Units	Description
Input: IRM T C0CP C1CP C2CP	m T C _{0Cm} C _{1Cm} C _{2Cm}	-- °C J/kg · °C J/kg · °C ² J/kg · °C ³	structure material index number ($1 \leq \text{IRM} \leq \text{NMATER} \leq 20$) structure temperature array of additive constants array of constants which multiply T array of constants which multiply T ²
Return: CPSOL1	C _p	J/kg · °C	specific heat capacity of structure material #IRM at temperature T. (CPSOL1 > 0)

Notes:

1. Replace "REAL" with "DOUBLE PRECISION" on short-word machines.
2. As currently coded, this FUNCTION is computed using $\text{CPSOL1} = (\text{C2CP}(\text{IRM}) * \text{T} + \text{C1CP}(\text{IRM})) * \text{T} + \text{C0CP}(\text{IRM})$, where C0CP, C1CP, and C2CP are the arrays input in NAMELIST DATA.

Table 9.5 Entry Point Description for FUNCTION XKSOL1

Entry Point:
 FUNCTION XKSOL1(IRM,T,C0K,C1K,C2K)
 INTEGER IRM
 REAL XKSOL1,T,C0K,C1K,C2K
 DIMENSION C0K(20),C1K(20),C2K(20)

Code	Text	Units	Description
Input: IRM T C0K C1K C2K	m T C _{0λm} C _{1λm} C _{2λm}	-- °C W/m · °C W/m · °C ² W/m · °C ³	structure material index number ($1 \leq \text{IRM} \leq \text{NMATER} \leq 20$) structure temperature array of additive constants array of constants which multiply T array of constants which multiply T ²
Return: XKSOL1	λ	W/m · °C	thermal conductivity of structure material #IRM at temperature T. (XKSOL1 > 0)

Notes:

1. Replace "REAL" with "DOUBLE PRECISION" on short-word machines.
2. As currently coded, this FUNCTION is computed using $\text{XKSOL1} = (\text{C2K}(\text{IRM}) * \text{T} + \text{C1K}(\text{IRM})) * \text{T} + \text{C0K}(\text{IRM})$ where C0K, C1K, and C2K are the arrays input in NAMELIST DATA.

9.3 Heat Transfer Coefficients

The heat transfer rate between a fluid cell and the associated surface of a solid (either a boundary or an immersed object) is generally computed using

$$Q = U_n A (T_s - T) , \quad (9.9)$$

where U_n is a heat transfer coefficient, A is the interfacial surface area, T_s is the temperature of the solid at the surface, and T is the fluid temperature. The user must input all heat transfer coefficients to be used in a calculation; each is assigned a positive index number, n . Then, for each fluid-solid thermal interaction, the user specifies which heat transfer coefficient, by index n , is to be used at the surface; the heat transfer coefficient index number is input using array element $IHTWAL(ns)$ for boundary surface ns (cf. Table 4.6) and using variable IHT in NAMELIST F in the Thermal Structure Prototype input block for a thermal structure (cf. Table 5.3). The same heat transfer coefficient index number may be used on more than one surface. There are three exceptions to the above generalities: (a) if heat transfer between boundary surface ns and the fluid is by conduction in the fluid (cf. Sec. 4.1.3), the user inputs $IHTWAL(ns)=0$ and the input described in the rest of this section is irrelevant for that surface; (b) when using the "duct wall" energy boundary condition (cf. Sec. 4.1.3), the heat transfer between boundary surface ns and the external surroundings is computed using a constant heat transfer coefficient whose value is input using $HSINK(ns)$ in NAMELIST $DATA$ (cf. Table 4.6) rather than a heat transfer coefficient index number being input; (c) when there is more than one material partition in a thermal structure element (cf. Sec. 5.2.3), the heat transfer across the gap between the two material regions is computed using a constant heat transfer coefficient whose value is input using $HGAP$ in NAMELIST M in the Thermal Structure Prototype input block (cf. Table 5.4) rather than a heat transfer coefficient index number being input.

The heat transfer coefficients are nominally computed from

$$U_n = \left(\frac{\lambda_l}{D_h} \right) Nu_n , \quad (9.10)$$

where λ_l is the laminar (molecular) thermal conductivity of the fluid and D_h is a characteristic

length for the surface (input using array element HYDWAL(ns) in NAMELIST DATA (cf. Table 4.4) for boundary surface ns and using variable HYD in NAMELIST F in the Thermal Structure Prototype input block (cf. Table 5.3) for a thermal structure). The term Nu_n is a Nusselt-number correlation, having the form

$$Nu_n = C_{1Nn} + C_{2Nn} Re^{C_{3Nn}} Pr^{C_{4Nn}} \quad (9.11)$$

The fluid Reynolds number is given by

$$Re = \frac{\rho D_h (\bar{u}^2 + \bar{v}^2 + \bar{w}^2)^{0.5}}{\mu_t} \quad (9.12)$$

where ρ is the fluid density, μ_t is the laminar (molecular) viscosity of the fluid, and the fluid speed is based on the spatial averages of the three velocity components, $\bar{u}$, $\bar{v}$, and $\bar{w}$. The fluid Prandtl number used in Eq. 9.11 is given by

$$Pr = \frac{\mu_t C_p}{\lambda_t} \quad (9.13)$$

where C_p is the specific heat capacity of the fluid.

In order to use heat transfer coefficients of the form given by Eqs. 9.10 through 9.11, the user uses heat transfer coefficient index numbers (i.e. IHTWAL(ns) and IHT) in the inclusive range from 1 through 20 and provides the values for the input variables shown in Table 9.6 in NAMELIST DATA. The value of NHEATC is the maximum of the values entered for IHTWAL(ns) and IHT which are less than or equal to 20. (Although not a requirement, the user will find it most convenient to use heat transfer coefficient index numbers in the range from 1 through 20 in a contiguous sequence beginning with 1; in this case, NHEATC will be the number of different coefficients used in this range.) The maximum value allowed for NHEATC is 20. For each heat transfer coefficient index number n used in the range from 1 through NHEATC, the user provides the values of the four constants C_{iNn} used in Eq. 9.11 using array elements HEATC1(n), HEATC2(n), HEATC3(n), and HEATC4(n). Since the heat transfer coefficient must always be positive, the user should enter $HEATC1(n) > 0$ to allow for situations for which the fluid velocity is zero.

Table 9.6 Heat Transfer Correlation Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
NHEATC	number of basic heat transfer coefficient correlations (i.e. the maximum of values entered for IHTWAL(N) in NAMELIST DATA and IHT in NAMELIST F of the Thermal Structure Prototype input block which are less than or equal to 20).	$0 \leq \text{NHEATC} \leq 20$
<p>Basic heat transfer coefficient correlation #n is assumed to be given by</p> $U_n = (\lambda_t/D_h) * \text{Nu}_n$ <p>where U_n is a heat transfer coefficient [$\text{W}/\text{m}^2 \cdot ^\circ\text{C}$], λ_t is the laminar (molecular) thermal conductivity of the fluid [$\text{W}/\text{m} \cdot ^\circ\text{C}$], D_h is a characteristic length [m], and Nu_n is a Nusselt number correlation given by</p> $\text{Nu}_n = C_{1\text{Nu}} + C_{2\text{Nu}} * \text{Re}^{C_{3\text{Nu}}} * \text{Pr}^{C_{4\text{Nu}}}$ <p>where the Reynolds and Prandtl numbers are</p> $\text{Re} = \rho * D_h * (\bar{u}^2 + \bar{v}^2 + \bar{w}^2) / \mu_t$ $\text{Pr} = \mu_t * C_p / \lambda_t$ <p>and ρ is the fluid density [kg/m^3], $\bar{u}$, $\bar{v}$, and $\bar{w}$ are average values of the three velocity components [m/s], μ_t is the laminar (molecular) viscosity of the fluid [$\text{Pa} \cdot \text{s}$], and C_p is the specific heat capacity of the fluid [$\text{J}/\text{kg} \cdot ^\circ\text{C}$].</p> <p>The next 4 variables must be input for each basic heat transfer correlation n.</p>		$1 \leq n \leq \text{NHEATC}$
HEATC1(n)	$C_{1\text{Nu}}$: additive constant [-] in Nusselt number correlation n.	$\text{HEATC1}(n) > 0$
HEATC2(n)	$C_{2\text{Nu}}$: multiplicative constant [-] in Nusselt number correlation n.	
HEATC3(n)	$C_{3\text{Nu}}$: Reynolds number exponent [-] in Nusselt number correlation n.	
HEATC4(n)	$C_{4\text{Nu}}$: Prandtl number exponent [-] in Nusselt number correlation n.	

Notes:

1. The user may use alternate forms for heat transfer coefficients by using correlation index numbers in the range 51 through 99 and supplying coding in FUNCTION HEATCU to calculate them.

If the heat transfer coefficient can not be adequately represented by the above general form, the user may supply additional specific forms, which are then referenced by values of IHTWAL(ns) and IHT in the inclusive range from 51 through 99. The user supplies these additional forms to COMMIX by providing (i.e. writing, compiling, and linking with the rest

of COMMIX (cf. Appendix C.2)) FUNCTION HEATCU. The coding for FUNCTION HEATCU must conform to the entry point description shown in Table 9.7. As shown, all local fluid and structure conditions, as well as the specific fluid cell location, are available as arguments. This FUNCTION is called as needed for each fluid/structure interface which uses a heat transfer coefficient index in the range from 51 through 99.

Table 9.7 Entry Point Description for FUNCTION HEATCU

Entry Point:

FUNCTION HEATCU(IH,M0,IJK,VEL,RL,TL,P,CP,RMU,XK,HYD,TW)

INTEGER IH,M0,IJK

REAL HEATCU,VEL,RL,TL,P,CP,RMU,XK,HYD,TW

Code	Text	Units	Description
Input:			
IH	n	--	correlation index number ($51 \leq IH \leq 99$)
M0	-	--	fluid cell number
IJK	i,j,k	--	fluid cell location indices (stored as $10000*i + 100*j + k$)
VEL	$(\bar{u}^2 + \bar{v}^2 + \bar{w}^2)^{0.5}$	m/s	fluid speed (3-D average)
RL	ρ	kg/m ³	fluid density
TL	T	°C	fluid temperature
P	P	Pa	fluid pressure
CP	CP	J/kg · °C	fluid specific heat capacity
RMU	μ_t	Pa · s	fluid laminar viscosity
XK	λ_t	W/m · °C	fluid laminar thermal conductivity
HYD	D_h	m	characteristic length
TW	T_s	°C	wall temperature
Return:			
HEATCU	U_n	W/m ² · °C	heat transfer coefficient from correlation #IH between a structure (temperature TW; characteristic length HYD) and the fluid (temperature TL; other properties RL, P, CP RMU, and XK; speed VEL) for fluid cell #M0 at fluid grid point location (i,j,k). (HEATCU > 0)

Notes:

1. Replace "REAL" with "DOUBLE PRECISION" on short-word machines.

10. HEXAGONAL FUEL ASSEMBLY

Several features are included in COMMIX specifically for modeling hexagonal assemblies of fuel pins arranged on a triangular pitch. This modeling is activated, primarily, by specifying a value for IGEOM which is greater than zero in NAMELIST GEOM. As shown in Table 10.1, the value of IGEOM is the number of fuel pins in the assembly. Having $IGEOM > 0$ simplifies the geometry input process and automatically invokes some frictional resistance due to the presence of the pins and the wire wraps.

The hexagonal fuel assembly option is only useful for modeling the complete cross-sectional area of a single fuel assembly having a uniform interior (i.e. uniform pin diameter, pin pitch, etc.). If the assembly differs from this uniform hexagonal geometry, the user must model the assembly under the $IGEOM=0$ geometry option and specify all of the geometry details manually; two examples of where the $IGEOM=0$ approach must be used are (1) modeling only a sector (e.g. 1/6 or 1/12) of a hexagonal assembly and (2) fuel assemblies which are not hexagonal. Several of the pin model options may be used for these nonstandard fuel assemblies; further details are given in Sec. 10.4.

10.1 Geometry

Section 3 of this report describes the procedure to be followed to input the geometry for a general problem in Cartesian ($IGEOM=0$) and cylindrical ($IGEOM=-1$) coordinates. Selection of $IGEOM > 0$ automates many parts of the geometry specification process, as described in the remainder of the present subsection. The user must understand the geometry concepts (e.g. fluid cells, boundary surfaces, and porosity) introduced in Sec. 3 before attempting to read the present subsection concerning the hexagonal geometry option.

The choice of $IGEOM > 0$ simplifies the user's work in preparing input to describe the fuel assembly geometry to the program. Under this option, the program will use a (x,y,z) Cartesian coordinate system for the fluid cells with the z axis being parallel to the length of the fuel pins. (The absolute orientation of the z axis is determined by the values input for the components of

Table 10.1 Hexagonal Fuel Assembly Variables in NAMELIST GEOM

Variable	Meaning/Value	Restrictions
IGEOM	number of fuel pins in the assembly	IGEOM=7,19,37,61, 91,127,169,217,271
CLADOD	$2r_p$: diameter of fuel pin [m].	
PITCH	distance between pin centers [m].	
WALLCL	clearance between pin and duct wall [m].	
IPART	type of partitioning for x-y planes; may be: 0: quarter-pin; 1: full-pin.	
IWIRE	wire wrap model selection; may be: 0: no wire wraps; 1: effect of wire wraps is uniform axially (smeared); no discrete positioning or swirl flow; 2: discrete positioning of wire wraps at each spatial location; activates cell-integrated wire force model; allows calculation of swirl flow.	IPART=0
WODIN	$d_{w,in}$: diameter of wire wrap in assembly interior [m].	IWIRE $\geq$ 1
WODOUT	$d_{w,out}$: diameter of wire wrap by duct wall [m].	IWIRE $\geq$ 1
WIREP	P_w : pitch of wire wrap [m]; WIREP>0 indicates counter-clockwise rotation of wrap when looking in the minus z direction; WIREP<0 is clockwise rotation.	IWIRE=2
ZATO	z_0 : axial position where wraps are on the plus x-axis relative to the pin center [m].	IWIRE=2

Table 10.1 Hexagonal Fuel Assembly Variables in NAMELIST GEOM (Cont'd)

Variable	Meaning/Value	Restrictions
KMAX	maximum number of cells in the z direction.	$1 \leq KMAX \leq 99$
DZ(K)	Δz : cell spacing [m] in the z direction.	$DX(K) > 0$. $1 \leq K \leq KMAX$
NM1	total number of cells.	
NL1	total number of surface elements.	
CWIREI	$C_{t,in}$: wire force multiplier [-] for wraps in the assembly interior.	IWIRE=2
CWIREO	$C_{t,out}$: wire force multiplier [-] for wraps next to the duct wall.	IWIRE=2

the gravitational acceleration vector (cf. Sec. 3.2); for example, the input GRAVX=0.0, GRAVY=0.0, and GRAVZ=-9.8 in NAMELIST DATA would specify a vertical assembly with the inlet at the bottom.) The eight boundary surfaces (one for each of the six faces of the hexagonal assembly wall, one for the inlet plane, and one for the outlet plane) needed to enclose the fluid cells are defined automatically by the program; the correspondences between boundary surface identification numbers and physical locations are shown in Table 10.2. Note that the first four of these boundary surfaces are IREGular -- the unit surface normal vector does not point in one of the three coordinate directions. The other four boundary surfaces are REGular. (The user is relieved of the responsibility of supplying the number of boundary surfaces (NSURF), the components of the unit normal vectors for these boundary surfaces (XNORML, YNORML, and ZNORML), and the Boundary Surface Identification input block.)

The division of the x-y plane into computational fluid cells is performed automatically by the program based on the values input by the user for a few key geometric quantities. Referring to Table 10.1, these are the fuel pin diameter (CLADOD), the distance between pin centers (PITCH), the clearance between the outermost row of pins and the assembly wall (WALLCL), and the type of planar partitioning (IPART). The two types of planar partitioning derive their

names from what fuel-pin fraction is associated with a fluid cell in the assembly interior; using a 19-pin assembly as an example, the difference between the two types of partitioning may be seen by comparing the quarter-pin partitioning shown in Fig. 10.1 with the full-pin partitioning shown in Fig. 10.2. With this basic geometry information, the program automatically sets the maximum number of cells (IMAX and JMAX) and the cell widths (DX(I) and DY(J)) for the x and y directions to the values shown in Table 10.3. The central pin is "pin ring #0" when determining the value for the term "# of pin rings" as used in Table 10.3. The program also sets the volume porosities (AL(I,J,K)) and the directional surface porosities (ALX, ALY, and ALZ(I,J,K)) to reflect the presence of the fuel pins. The specific values assigned to interior cells and cell faces are shown in Table 10.3; note that the x and y directional surface porosities for quarter-pin partitioning depend on the specific location of the cell face relative to the fuel pin (cf. Fig. 10.1). A similar procedure is used to assign porosity values for cells near the wall; this procedure recognizes that the geometry is different from the assembly interior, that some cells are IREGular, and that some cells and cell faces are not intersected by any fuel pins.

Table 10.2 Boundary Surface Assignments for Hexagonal Fuel Assembly
Geometry Option (IGEOM > 0)

Surface Number	Type	Unit Surface-Normal Vector			Physical Location
		n_x	n_y	n_z	
1	IREG	$+\sqrt{3}/2$	$+1/2$	0	lower-left diagonal flat
2	IREG	$+\sqrt{3}/2$	$-1/2$	0	upper-left diagonal flat
3	IREG	$-\sqrt{3}/2$	$+1/2$	0	lower-right diagonal flat
4	IREG	$-\sqrt{3}/2$	$-1/2$	0	upper-right diagonal flat
5	REG	0	+1	0	lower flat (on x axis)
6	REG	0	-1	0	upper flat
7	REG	0	0	+1	inlet plane (z=0)
8	REG	0	0	-1	exit plane

Fig. 10.1 Quarter-Pin Partitioning of a 19-Pin Fuel Assembly in a Hexagonal Duct

Fig. 10.2 Full-Pin Partitioning of a 19-Pin Fuel Assembly in a Hexagonal Duct

Table 10.3 Other Geometry Variable Values Automatically Set for Hexagonal Fuel Assembly Option (IGEOM > 0)

Variable	Type of Partitioning in x-y Plane	
	Full Pin (IPART=1)	Quarter Pin (IPART=0)
IMAX	(# rings of pins + 1)*2	((# rings of pins)*2 + 1)*2
JMAX	IMAX	same as Full Pin
NSURF	8	same as Full Pin
DX(I)	for I=1 and IMAX: (0.5*CLADOD+WALLCL)/cos($\pi/6$) for I=2,...,IMAX-1: PITCH	same as Full Pin 0.5*PITCH
DY(J)	for J=1 and JMAX: 0.5*CLADOD+WALLCL for J=2,...,JMAX-1: PITCH*cos($\pi/6$)	same as Full Pin 0.5*PITCH*cos($\pi/6$)
Nonunity porosities are set for interior cells and cell faces to reflect presence of the pins as follows (I=1,2,...,IMAX, J=1,2,...,JMAX, and K=1,2,...,KMAX unless noted otherwise):		
AL(I,J,K)	$1 - \frac{(\pi/4)*(CLADOD)^2}{DX(I)*DY(J)}$	$1 - \frac{(\pi/16)*(CLADOD)^2}{DX(I)*DY(J)}$
ALX(I,J,K)	$1 - \frac{CLADOD/2}{DY(J)}$	for cell faces in Note 1: $1 - \frac{CLADOD/2}{DY(J)}$
ALY(I,J,K)	$1 - \frac{CLADOD}{DX(I)}$	for J=1,3,...,JMAX-1: $1 - \frac{CLADOD/2}{DX(I)}$
ALZ(I,J,K)	$1 - \frac{(\pi/4)*(CLADOD)^2}{DX(I)*DY(J)}$	$1 - \frac{(\pi/16)*(CLADOD)^2}{DX(I)*DY(J)}$

Notes:

- For quarter-pin partitioning, ALX(I,J,K) is set to the value indicated above for the following cell faces:
IGEOM=7,37,91,169,271:
I=2,4,...,IMAX, J=1,2,5,6,...,JMAX-1,JMAX, K=1,2,...,KMAX
I=1,3,...,IMAX-1, J=3,4,7,8,...,JMAX-3,JMAX-2,K=1,2,...,KMAX
IGEOM=19,61,127,217:
I=1,3,...,IMAX-1, J=1,2,5,6,...,JMAX-1,JMAX, K=1,2,...,KMAX
I=2,4,...,IMAX, J=3,4,7,8,...,JMAX-3,JMAX-2, K=1,2,...,KMAX.
- Porosities of cells and cell faces adjacent to boundary surfaces are first adjusted to reflect the presence of the IREGular boundary surfaces and then adjusted, if needed, to reflect the presence of the pins.
- Additional adjustment of porosity values to reflect the presence of wire wraps is discussed in the text.
- The following porosity values are not used in the code: ALX(IMAX,J,K), ALY(I,JMAX,K), and ALZ(I,J,KMAX).

Many hexagonal fuel assemblies use spiralling wires on the pins ("wire wraps") to maintain proper pin-to-pin spacing. The value input for IWIRE (cf. Table 10.1) determines how these wire wraps are treated in COMMIX: (a) IWIRE=1 selects the smeared-wrap option, in which the effect of the wraps is distributed uniformly over the x-y plane and the length of the assembly and (b) IWIRE=2 selects the cell-integrated-wrap option, in which the effect of the wraps is constant within any given cell but varies from cell to cell (both within the x-y plane and from one axial level to another) depending on the actual location of each wire wrap. For both of these options, the user must specify the wrap's diameter (using WODIN for wraps in the assembly interior and WODOUT for wraps adjacent to the assembly wall). For the cell-integrated-wrap model (which may only be selected in conjunction with quarter-pin partitioning), the user must also input the pitch of the wrap (WIREP) and a reference axial position (ZAT0) at which the wraps are aligned with the positive x axis. Note that the sign of WIREP is used to indicate whether the wrap's spiral direction is clockwise or counterclockwise. The reference axial position specified using ZAT0 is relative to the axial grid being used in the calculation (see next paragraph) and may be any position at which a vector drawn in the x-y plane from the center of a pin to the center of its wire wrap points in the direction of the +x axis (cf. axis locations in Fig. 10.1). The program uses the above information to adjust the porosities to account for the presence of the wire wraps using Eqs. A.1 through A.4 of Ref. 1; the first term on the right-hand side of these equations is the porosity after having been corrected for the presence of the pins; the second term is the adjustment for the presence of the wire wraps. For the cell-integrated-wrap option, the limits in the integrals (z_1 and z_2) span one axial level and the equations are applied at each axial level; for the smeared-wrap option, the integrals are only evaluated for one axial level and the resulting values are used for all axial levels. (Additional user input associated with the forces computed for the IWIRE=2 option is discussed in Sec. 10.2.) If the wire wraps are to be ignored or there are no wire wraps, input IWIRE=0 in NAMELIST GEOM. (There is no automatic modeling for grid spacers, which are used in some fuel assemblies for pin positioning; the user must introduce the geometry effects by specifying nonunity porosities (cf. Sec. 3.5) and the frictional effects by defining force structures (cf. Sec. 6).)

The division of the assembly length into mesh cells must be specified by the user. Referring

to Table 10.1, the number of mesh cells in the z direction is specified using input variable KMAX, and the mesh cell spacings in the z direction are input using array DZ(K) (for K=1 to KMAX, where K=1 is the axial level adjacent to boundary surface number 7). When using the cell-integrated wire-wrap model, the user will want to consider the value of wrap pitch (WIREP) in selecting the values of axial mesh spacing: if the axial mesh spacing is too large relative to the wrap pitch (i.e. $DZ(Z) > WIREP/4$ when IWIRE=2), the effect of the wire wrap spiralling will not be fully representable in the modeling. As for other geometries, the user must input the number of fluid cells and the number of surface elements using variables NM1 and NL1, respectively, in NAMELIST GEOM. The user must also provide the grid orientation (cf. Table 3.2) and the fluid assignment (cf. Tables 3.3, 3.4, and 3.8) information described in Section 3 of this report.

10.2 Forces from Fuel Pins and Wire Wraps

Table 10.4 shows the input variables in NAMELIST DATA associated with the hexagonal fuel assembly model. These input variables provide additional information for the fuel pin and the wire wrap models.

A nonzero value for input variable IFROD indicates that one or more of the special fuel pin model features are to be used. These special models are for the drag force introduced by the presence of the fuel pins and a simplified description of the heat source provided by the fuel pins. Each of these models requires additional user input, as described in this section and the next section. (The one-dimensional-based initialization selected by IFROD=2 is discussed in Sec. 10.4.) When IFROD=0, the special fuel pin models are not used.

The special fuel-pin drag-force model is selected by IDRAG > 0. The equations describing this model are given in Sec. A.3 of Ref. 1. If this model is selected, then input variables CDRAGX, CDRAGY, and CDRAGZ (cf. Table 10.4) may be used to override the default values of the multipliers which are applied to the forces obtained from the built-in correlations. (If this model is not selected, the drag force due to the fuel pins should be described using the force structure model (cf. Sec. 6).)

Table 10.4 Hexagonal Fuel Assembly Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
IFROD	fuel pin model selection; may be: 0: model not used; omit NAMELIST INPUTQ; 1: model is used; include NAMELIST INPUTQ; 2: model is used and one-dimensional-based fluid initialization is performed; include NAMELIST INPUTQ.	ISTATE=0 or 2 ISTATE=0 IFHTX=1
IDRAG	fuel pin drag force model selection; may be: 0: model not used; any drag forces desired must be described using the Force Structure Model (cf. Sec. 6); 1: nominal model used; 2: same as IDRAG=1 except the nominal x - and y -direction forces are multiplied by γ/γ_x or γ/γ_y ; should generally be selected when IWIRE=2.	IFROD > 0 IFROD > 0
CDRAGX	$C_{t,x}$: x -direction fuel pin drag force multiplier [-] (in Eq. A.17 of Ref. 1).	IDRAG > 0
CDRAGY	$C_{t,y}$: y -direction fuel pin drag force multiplier [-] (in Eq. A.17 of Ref. 1).	IDRAG > 0
CDRAGZ	$C_{t,z}$: z -direction fuel pin drag force multiplier [-] (in Eq. A.17 of Ref. 1).	IDRAG > 0
CWIREX	C_x : x -direction wire force multiplier [-] (in Eq. A.7 of Ref. 1).	IWIRE=2
CWIREY	C_y : y -direction wire force multiplier [-] (in Eq. A.7 of Ref. 1).	IWIRE=2
CWIREZ	C_z : z -direction wire force multiplier [-] (in Eq. A.7 of Ref. 1).	IWIRE=2

Selection of the discrete-positioning wire-wrap model (IWIRE=2 in NAMELIST GEOM) activates a special model which causes the changing position of the wire wrap to exert a force on the fluid. The equations describing this model are given in Sec. A.2.3.2 of Ref. 1. When this wire-wrap model is used, the user may override the default values of the multipliers which are applied to the forces obtained from the built-in correlations. The overall force multiplier is

the product of two multipliers: (1) a multiplier based on the spatial direction of the force, which is given by either CWIREX (C_x), CWIREY (C_y), or CWIREZ (C_z) as shown in Table 10.4 and (2) a multiplier (C_i) based on whether the fluid cell is in the bundle interior or adjacent to the assembly wall, which is given by either CWIREI or CWIREO as shown in Table 10.1.

10.3 Heat Generation

There are three methods by which the heat source provided by the fuel pins may be described. The first of these methods is use of the thermal structure model to describe the fuel pins; this is the most detailed method and is the only method under which the temperature distribution within the fuel pins is calculated; the use of the thermal structure model is discussed in Sec. 5; one or more thermal structure prototypes may be needed to describe all of the fuel pins. The second of these methods is specification of the heat source as an equivalent volumetric heat source directly in the coolant; as described in Sec. 4.1.5, the way in which this is accomplished depends on whether there are any thermal structures used anywhere in the calculation: if ISTRUC=0 in NAMELIST GEOM, use 'QSOU' records in the Internal Cell Initialization input block (Table 4.10) and input variable NOFQT in NAMELIST DATA (Table 4.6); if ISTRUC>0, use input variables QSCOO, QIN, and QK in NAMELIST INPUTQ (Table 4.7). The third method is the simplified fuel pin model, which is described below.

A simplified model is provided for describing the heat source provided to the fluid by the fuel pins. In this model, the thermal inertia of the fuel pins is ignored and the details of the temperature distributions within the fuel pins are not modeled; the heat transfer rate (Q) from the portion of the fuel pin(s) in contact with the fluid cell having spatial location indices (i,j,k) is computed from

$$Q(i,j,k) = QIJ(i,j)*QK(k)*(2\pi r_p \Delta z_k)*PINF(i,j)*QFLUX*f_{NOFQT}(t), \quad (10.1)$$

where QIJ and QK are power distribution functions in the x-y plane and in the z direction, respectively, $2r_p$ is the pin diameter (CLADOD), Δz_k is the length in the z direction of the fluid

cells at axial location k (array element $DZ(k)$), $PINF$ is the number (or fraction) of fuel pins contacting fluid cells at location (i,j) , $QFLUX$ is the average heat flux, and $f_{NOFQT}(t)$ is transient function number $NOFQT$; these terms are discussed in the next several paragraphs. There is no specific user input flag which activates use of this model; instead, the heat source to coolant will be calculated from Eq. 10.1 if there are no thermal structures used anywhere in the calculation ($ISTRUC=0$); conversely, if thermal structures are used anywhere in the calculation ($ISTRUC>0$), then they must also be used to describe the fuel pins (and, thus, Eq. 10.1 is not used).

Table 10.5 shows the input variables in NAMELIST INPUTQ associated with the hexagonal fuel assembly model. Most of this input is associated with fuel-pin heat generation. This block of input is only read if (1) $IFROD>0$ (cf. Table 10.4) and (2) $ISTATE=0$ or $ISTATE=2$. Normally the items in this namelist are specified completely at the beginning of a steady-state calculation ($ISTATE=0$) and not changed later when beginning a transient calculation from a steady-state calculation (i.e. specify an empty namelist (cf. Sec. 2) at $ISTATE=2$).

In Eq. 10.1, the average heat flux at the surface of the fuel pins is specified using input variable $QFLUX$. If this average heat flux is a function of time, then enter a nonzero value for $NOFQT$ and define a transient function having this index number (cf. Sec. 4.1.4); if the heat flux is constant in time, input either (a) $NOFQT=0$ (in which case the term " $f_{NOFQT}(t)$ " will be replaced by "1.0") or (b) $NOFQT>0$ and define transient function number $NOFQT$ as having a constant value.

The type of axial power distribution is indicated by the value of input variable IQ . There are four choices: a user-described shape ($IQ=0$) or one of the three internally generated shapes ($IQ>0$). For all shape choices, the user specifies the lowest and highest axial level having heat generation using input variables $KLHS$ and $KHHS$, respectively. For $IQ=0$, the user specifies the axial power distribution by axial node using input array QK . For $IQ=1$, the program fills array QK with a distribution having a "cosine" shape with the peak power midway between axial levels $KLHS$ and $KHHS$. The other two choices for IQ also result in cosine-shaped distributions but the peak is skewed toward node $KHHS$ (for $IQ=2$) or node $KLHS$ ($IQ=3$). For $IQ>0$, the

Table 10.5 Hexagonal Fuel Assembly Variables in NAMELIST INPUTQ¹

Variable	Meaning/Value	Restrictions
QFLUX	average heat flux [W/m ²] at pin surface.	ISTRUC=0
NOFQT	transient function number by which to multiply heat source given by QFLUX or QSOUR(I,J,K).	ISTATE=2 0 ≤ NOFQT ≤ 25
IQ	type of axial power distribution (array QK); may be: 0: user-specified; input array QK; 1: cosine shape with peak at mid-height; 2: cosine shape with peak skewed toward top; 3: cosine shape with peak skewed toward bottom.	
KLHS	lowest heated K plane.	1 ≤ KLHS ≤ KHHS
KHHS	highest heated K plane.	KLHS ≤ KHHS ≤ KMAX See note 4.
FNZ	ratio [-] of peak to average in axial power distribution.	IQ > 0
QK(K)	relative power [-] in axial node K; the values of QK(K) must be positive for KLHS ≤ K ≤ KHHS.	IQ = 0 1 ≤ K ≤ KMAX See note 2.
QIN(IND)	relative power [-] for cell position (I,J), where IND = I + IMAX*(J-1).	1 ≤ IND ≤ 100 IMAX ≤ 10 JMAX ≤ 10 See note 2.
CLADOD	2r _p : diameter [m] of fuel pin. (Do not need to reinput here if specified in NAMELIST GEOM.)	QFLUX ≠ 0 IDRAG > 0
PITCHX	P _x : x-direction component of fuel pin pitch [m].	IDRAG > 0 See note 3.
PITCHY	P _y : y-direction component of fuel pin pitch [m].	IDRAG > 0 See note 3.
IJTYPE(IND)	cell-type identifier for cell position (I,J), where IND = I + IMAX*(J-1); used in connection with the next two arrays to override default values set by the program for IGEOM > 0.	1 ≤ IND ≤ 100 IJTYPE ≤ 50 See note 3.
PINAF(IJ)	fuel pin fraction [-] in cell type IJ; if any values are to be input, you must enter values for all cell types (not just for those cell types for which default is to be overridden).	1 ≤ IJ ≤ max(IJTYPE(IND)) QFLUX ≠ 0 See note 3.

Table 10.5 Hexagonal Fuel Assembly Variables in NAMELIST INPUTQ (Cont'd)

Variable	Meaning/Value	Restrictions
WETLN(IJ)	wetted perimeter [m] in cell type IJ. (If specifying values here, you should probably also be specifying revised fluid volumes and flow areas, which is accomplished using 'AL ', 'ALX ', 'ALY ', and 'ALZ ' records in the Internal Cell Initialization input block (see Sec. 3.5).)	$1 \leq IJ \leq \max(IJTYPE(IND))$ IDRAG > 0 See note 3.

Notes:

1. Only read if IFROD > 0 in NAMELIST DATA.
2. Arrays QIN and QK are used as shape multipliers for duct wall, internal fluid, thermal structures, and hexagonal fuel assembly heat sources. The values input will be normalized by the program.
3. PITCHX, PITCHY, IJTYPE, PINAF, and WETLN are set internally by the program for IGEOM > 0; the user should not usually override these internally-set values.
4. Current array dimensions require KHHS ≤ 49 when IQ > 0.

ratio of the peak to the average in this axial distribution must be specified using input variable FNZ. The shape of the power distribution in the other two directions is input by the user using array QIN, which is used by the program to fill array QIJ (i.e. $QIJ(i,j) = QIN(i + IMAX*(j - 1))$).

The number of fuel pins contacting each fluid cell (PINF in Eq. 10.1) is automatically set by the program when IGEOM > 0; the values depend on cell location and type of x-y plane partitioning (IPART in Table 10.1); for quarter-pin partitioning (selected by IPART=0 and illustrated in Fig. 10.1), the values of PINF are 0.25 for all cells except for a few cells adjacent to the assembly wall (for which the values are 0.0); for full-pin partitioning (selected by IPART=1 and illustrated in Fig. 10.2), the values of PINF are 1.0 for interior cells and 0.0, 0.25, 0.5, or 0.75 (depending on exact location) for cells adjacent to the assembly wall. For nonstandard assembly geometries, the user may input this information using arrays IJTYPE and PINAF; further information on nonstandard assembly geometries is discussed in Sec. 10.4.

10.4 Other Considerations

As is the case for all calculations, the user must provide boundary and initial conditions (cf. Sec. 4), fluid properties (cf. Sec. 9.1), and auxiliary input (cf. Sec. 12). The user may also use the other optional modeling features in the program described elsewhere in this report (cf. Secs. 5–11).

For a regular hexagonal fuel assembly ($IGEOM > 0$), the boundary surfaces are numbered as shown in Table 10.2. The user must provide a full set of boundary conditions for these surfaces. Typical boundary condition choices are shown in Table 10.6, where u_{in} is the inlet velocity, P_{in} is the inlet pressure, P_{out} is the exit pressure, and T_{in} is the inlet temperature. A complete description of the input variables associated with boundary conditions is given in Sec. 4.1.

Table 10.6 Typical Boundary Conditions for Hexagonal Fuel Assemblies

Surface Number (n)	Physical Meaning	Typical Input Values in NAMELIST DATA
1-6	walls	KFLOW(n)=1, VELOC(n)=0, KPRES(n)=0 KTEMP(n)=500+nf
7	inlet	KFLOW(n)=1 (or 100+nf), VELOC(n)= u_{in} , KPRES(n)=0 - or - KFLOW(n)=-5, KPRES(n)=1 (or 100+nf), PRES(n)= P_{in} - plus - KTEMP(n)=1 (or 100+nf), TEMP(n)= T_{in}
8	exit	KFLOW(n)=-5, KPRES(n)=1 (or 100+nf), PRES(n)= P_{out} KTEMP(n)=400

As is the case for other types of geometries, initial conditions must be provided for the calculation. The several initialization processes and their associated input variables are described in Sec. 4.2. An additional one-dimensional-based initialization option (described in Sec. 10.2 of Ref. 1) is useful for fuel assemblies; this option is selected by inputting IFROD=2 in NAMELIST DATA (cf. Table 10.4). The choice of IFROD=2 should only be input at the

beginning of a steady-state calculation (ISTATE=0), since the values set by this one-dimensional-based initialization option override the values set by all of the other initialization processes described in Sec. 4.2 (including those values read from the restart file); the user must specify IFROD<2 in continuation runs (ISTATE>0) to avoid repeating the one-dimensional-based initialization.

Two of the pin modeling features developed for standard hexagonal fuel assemblies (IGEOM>0) may also be invoked when modeling nonstandard fuel assemblies (IGEOM=0): (1) the drag-force model and (2) the simplified heat-source model. The drag-force model is described in Sec. 10.2; its use requires IFROD>0 and IDRAG>0 (cf. Table 10.4); the user must provide values in NAMELIST INPUTQ (cf. Table 10.5) for the fuel pin diameter (CLADOD), the x and y components of the fuel pin pitch (PITCHX and PITCHY), and the wetted perimeter distribution (arrays IJTYPE and WETLN); the array containing the wetted perimeter for the cells having spatial location indices (i,j) will be filled by the program according to

$$WETLEN(i,j) = WETLN(IJTYPE(i+IMAX*(j-1))) \quad (10.2)$$

using the values input in arrays IJTYPE and WETLN; the user may override the default drag-force multipliers by inputting values for CDRAGX, CDRAGY, and CDRAGZ in NAMELIST DATA (cf. Table 10.4). The simplified heat-source model is described in Sec. 10.3; its use requires ISTRUC=0 in NAMELIST GEOM and IFROD>0 in NAMELIST DATA; in addition to the input values described in Sec. 10.3, the user must provide values in NAMELIST INPUTQ for the fuel pin diameter (CLADOD) and the fuel pin fraction distribution (arrays IJTYPE and PINAF); the array used in Eq. 10.1 which contains the fuel pin fraction for the cells having spatial location indices (i,j) will be filled by the program according to

$$PINF(i,j) = PINAF(IJTYPE(i+IMAX*(j-1))) \quad (10.3)$$

using the values input in arrays IJTYPE and PINAF.

As a final note, the size of certain arrays in the program places a limit on the fuel assembly size which can be modeled when certain options are selected. In particular, the values of IMAX and JMAX must not exceed 10 when any of the following models are to be used: (1) the fuel-pin drag-force model ($IDRAG > 0$) and (2) the simplified fuel-pin heat-source model ($ISTRUC = 0$ and $IFROD > 0$). This effectively limits the hexagonal assembly size to 61 pins when full-pin partitioning ($IPART = 1$) is used and 19 pins when quarter-pin partitioning ($IPART = 0$) is used. Although this places a rather severe restriction on assembly size, the array sizes have been kept small to minimize core storage requirements since the fuel assembly model options are not frequently used. (Larger assembly sizes can be modeled if the dimensions of the following arrays are increased in the source coding: WETLEN and QIJ in common block R4ARY and IJTYPE, PINF, and QIN in routine QGENER; contact the person who maintains the program at your site or one of the code authors for assistance.)

11. MASS REBALANCING

Mass rebalancing is a scheme used to accelerate the convergence of the inner iteration process used to solve the pressure equation system. In the mass rebalancing scheme, grid cells are grouped together to form rebalancing regions. A pressure correction equation is derived for each rebalancing region by summing the pressure equations for all grid cells in the rebalancing region. The following two conditions are applied in forming and working with the summations: (1) A uniform pressure correction is applied to all cells within a rebalancing region. (2) The pressure correction for each rebalancing region is determined such that the sum of the mass residuals over all cells in that region is zero. Further details on the mass rebalancing technique are described in Appendix B of Ref. 1.

Selection of mass rebalancing is optional. Mass rebalancing should only be invoked when the pressure equation system is solved using one of the successive overrelaxation solution methods (selected by ISYMCH=3 or 4 as described in Sec. 12.2). Mass rebalancing should not be invoked when solving the pressure equation system using the conjugate gradient method (selected by ISYMCH=5 as described in Sec. 12.2), which is the recommended method for solving the pressure equation system. Mass rebalancing has not been tested in multifluid problems (indicated by IFHTX > 1 as described in Sec. 3.3).

There are two ways in which cells may be grouped to form rebalancing regions: user-specified and plane-by-plane. The methods may be used individually or may be combined. The input variables associated with mass rebalancing are described in the following two subsections.

11.1 User-Specified Rebalancing

For user-specified rebalancing, the user specifies how cells are grouped together to form rebalancing regions. The general goal is to group together cells which are expected to have similar pressures. The rebalancing regions must be defined such that all cells in rebalancing region n are physically adjacent only to other cells in rebalancing region n and to cells in rebalancing regions $n-1$ and/or $n+1$. Rebalancing region 1 would typically be at the inlet of the

problem geometry and higher number rebalancing regions would be defined sequentially as one moves from the inlet toward the outlet. The user may define up to 99 rebalancing regions. An example of user-specified rebalancing regions is shown in Fig. 11.1.

Fig. 11.1 Example of User-Specified Mass Rebalancing Regions

The input associated with user-specified rebalancing is in three sections of the card-image input file: NAMELIST GEOM, NAMELIST DATA, and the Rebalancing Region Location block. The specific input details for each of these sections are shown in Tables 11.1 through

Table 11.1 User-Specified Rebalancing Variables in NAMELIST GEOM

Variable	Meaning/Value	Restrictions
IFREB	<p>indicates use of user-specified rebalancing regions; may be:</p> <p>=0: none used; omit Rebalancing Region Location input block;</p> <p>>0: used; the value input must be at least as large as the sum of the number of cells in all rebalancing regions plus the number of cell faces used to form all rebalancing surfaces; include additional input in NAMELIST DATA and the Rebalancing Region Location input block.</p>	<p>ISTATE=0 or 2</p> <p>ISYMCH<5</p>

Table 11.2 User-Specified Rebalancing Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
NEWREB	<p>indicator of user-specified rebalancing region definition input in this computer run: denotes presence of Rebalancing Region Location input block in this file; may be:</p> <p>0: omitted;</p> <p>1: included.</p>	<p>ISTATE=0 or 2</p> <p>IFREB>0</p>
NREBRT	number of user-specified rebalancing regions.	<p>NEWREB=1</p> <p>NREBRT ≤ 99</p>
NREBM(NR)	number of fluid cells in user-specified rebalancing region NR.	<p>NEWREB=1</p> <p>1 ≤ NR ≤ NREBRT</p>
NREBX(NR)	number of x (r) direction cell faces separating user-specified rebalancing regions NR and NR + 1.	<p>NEWREB=1</p> <p>1 ≤ NR ≤ NREBRT</p>
NREBY(NR)	number of y (θ) direction cell faces separating user-specified rebalancing regions NR and NR + 1.	<p>NEWREB=1</p> <p>1 ≤ NR ≤ NREBRT</p>
NREBZ(NR)	number of z direction cell faces separating user-specified rebalancing regions NR and NR + 1	<p>NEWREB=1</p> <p>1 ≤ NR ≤ NREBRT</p>
IREBIT	frequency at which rebalancing is performed, i.e. before every IREBIT-th inner iteration.	IREBIT > 0

Table 11.3 Rebalancing Region Location Input Block Record Structure

Position	Format	Variable	Meaning/Value	Restrictions
1-4	A4	NAME	card identifier; may be: 'REBM': the cells specified are part of rebalancing region "NUM"; 'REBX': the x (or r) direction cell faces specified are part of rebalancing surface "NUM"; 'REBY': the y (or θ) direction cell faces specified are part of rebalancing surface "NUM"; 'REBZ': the z direction cell faces specified are part of rebalancing surface "NUM"; 'END ': terminates this input block; record positions 5-32 should be blank.	
5-8	I4	NUM	rebalancing region or surface index number.	$1 \leq \text{NUM} \leq \text{NREBRT}$
9-12	I4	IB	beginning I-range index.	$1 \leq \text{IB} \leq \text{IE}$
13-16	I4	IE	ending I-range index.	$\text{IB} \leq \text{IE} \leq \text{IMAX}$
17-20	I4	JB	beginning J-range index.	$1 \leq \text{JB} \leq \text{JE}$
21-24	I4	JE	ending J-range index.	$\text{JB} \leq \text{JE} \leq \text{JMAX}$
25-28	I4	KB	beginning K-range index.	$1 \leq \text{KB} \leq \text{KE}$
29-32	I4	KE	ending K-range index.	$\text{KB} \leq \text{KE} \leq \text{KMAX}$
33-80	n.a.	n.a.	user comment (ignored).	Optional.

Notes:

1. This input block is present if, and only if, $\text{IFREB} > 0$ and $\text{NEWREB} = 1$, which is only permitted if $\text{ISTATE} = 0$ or $= 2$.
2. These records result in assigning the fluid cells in spatial positions (I,J,K) within the three-dimensional region delimited by $\text{IB} \leq \text{I} \leq \text{IE}$, $\text{JB} \leq \text{J} \leq \text{JE}$, and $\text{KB} \leq \text{K} \leq \text{KE}$ within the system volume to be part of rebalancing region "NUM" for "NAME" = 'REBM' and define the cell faces which form rebalancing surface "NUM" (which separates rebalancing region "NUM" from rebalancing region "NUM" + 1) for "NAME" = 'REBX', 'REBY', and 'REBZ'.
3. Use as many 'REBM', 'REBX', 'REBY', and 'REBZ' records as needed to describe the rebalancing regions and surfaces; the order of the records is irrelevant, except the 'END ' record only occurs once and is last; a cell or cell face must not be specified more than once in these records.
4. For 'REBM', the indices (I,J,K) refer to the cell-center grid points; the cells must actually exist (i.e. contain fluid); the number of cells specified for rebalancing region "NUM" must exactly match the value input in array location $\text{NREBRT}(\text{NUM})$ in NAMELIST DATA.

Table 11.3 Rebalancing Region Location Input Block Record Structure (Cont'd)

5. For 'REBX', the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I+1,J,K); the number of cell faces specified for rebalancing surface "NUM" must exactly match the value input in array location NREBX(NUM) in NAMELIST DATA.
6. For 'REBY', the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I,J+1,K); the number of cell faces specified for rebalancing surface "NUM" must exactly match the value input in array location NREBY(NUM) in NAMELIST DATA.
7. For 'REBZ', the indices (I,J,K) refer to the cell face between cells (I,J,K) and (I,J,K+1); the number of cell faces specified for rebalancing surface "NUM" must exactly match the value input in array location NREBZ(NUM) in NAMELIST DATA.
8. For 'REBX', 'REBY', and 'REBZ', cell faces having zero surface porosity should not be specified as forming part of a rebalancing surface.

11.3. As shown in Table 11.1, user-specified rebalancing is selected by inputting $IFREB > 0$ in NAMELIST GEOM; the specific value to be input will be discussed later in this subsection. User-specified rebalancing is not used when $IFREB = 0$. The decision to use user-specified rebalancing must be made at the beginning of a steady-state ($ISTATE = 0$) or beginning of a transient ($ISTATE = 2$) calculation.

The user must specify (a) which cells are in each rebalancing region and (b) which cell faces separate rebalancing region n from rebalancing region $n+1$ (which is identified as rebalancing surface n). This is accomplished using a set of fixed-format records in the Rebalancing Region Location input block in the card-image input file. The record structure for this input block is shown in Table 11.3. The first field in these records indicates the record type: the 'REBM' records specify which cells are in each rebalancing region; the 'REBX', 'REBY', and 'REBZ' records specify which x- (or r-), y- (or θ -), and z-direction cell faces, respectively, form each rebalancing surface. The second field in these records is the rebalancing region or rebalancing surface index number. The third through eighth fields in these records specify a range of cells or cell faces (by spatial location index) which are to be part of the indicated rebalancing region or surface. Use as many of each record type as needed to describe the rebalancing regions and surfaces. After all regions and surfaces are defined, an 'END' record must be present to terminate the Rebalancing Region Location input block. For the example shown in Fig. 11.1, the records needed to define the first three rebalancing regions and surfaces are shown in Table 11.4 (toward the bottom), assuming that the rebalancing regions and surfaces extend uniformly through the depth of the geometry for $J = 1$ to 10.

The Rebalancing Region Location input block is only read if NEWREB=1 in NAMELIST DATA. As shown in Table 11.2, this is only permitted at the beginning of a steady-state or transient calculation (ISTATE=0 or 2). Runs which continue a calculation in the same mode (ISTATE=1 or 3) are not allowed to newly specify or redefine the user-specified rebalancing regions; these continuation runs must have NEWREB=0 in NAMELIST DATA and omit the Rebalancing Region Location input block from the card-image input file; if user-specified rebalancing was in effect during the previous run, the rebalancing information will be read from the restart file. If user-specified rebalancing was selected at the beginning of a steady-state calculation (ISTATE=0), this information does not need to be respecified at the beginning of a transient run (ISTATE=2) unless the user wishes to make changes; to change the user-specified rebalancing at the beginning of a transient run, the user must input the Rebalancing Region Location input block in its entirety, not just those portions which are to be changed.

The additional input required in NAMELIST DATA for user-specified rebalancing is shown in Table 11.2. The number of rebalancing regions (shown as N in Fig. 11.1) is input using variable NREBRT. The number of cells in rebalancing region NR is entered in array element NREBM(NR), for NR=1,2,...,NREBRT; each entry is the sum of the number of cells specified in the 'REBM' records for that region. Similarly, the number of x- (or r-), y-(or θ -), and z-direction cell faces forming rebalancing surface NR are entered, respectively, in array elements NREBX(NR), NREBY(NR), and NREBZ(NR), for NR=1,2,...,NREBRT; each entry is the sum of the number of cell faces specified in the 'REBX', 'REBY', or 'REBZ' records, respectively, for that surface. For the example shown in Fig. 11.1, the entries in arrays NREBM, NREBX, NREBY, and NREBZ for the first three rebalancing regions and surfaces are shown in Table 11.4, assuming that the rebalancing regions and surfaces extend uniformly through the depth of the geometry for J=1,2,...,10. The value to be entered for IFREB in NAMELIST GEOM must be at least as large as the sum of the number of cells in all rebalancing regions plus the sum of all cell faces on all rebalancing surfaces; symbolically, this is expressed as

$$IFREB \geq \sum_{NR=1}^{NREBRT} [NREBM(NR) + NREBX(NR) + NREBY(NR) + NREBZ(NR)] \quad (11.1)$$

The information described in this paragraph should only be specified when NEWREB=1 and

Table 11.4 Portions of Input for User-Specified Rebalancing Example
Shown in Figure 11.1

```

&GEOM
  IFEB=...,
&END

&DATA
  NEWREB=1, NREBRT=..., IREBIT=...,
  NREBM= 50,110,110,...,
  NREBX= 10, 10,  0,...,
  NREBY=  0,  0,  0,...,
  NREBZ= 50, 90, 90,...,
&END

  REBM   1   1   5   1  10   1   1   Rebal. Region 1
  REBX   1   5   5   1  10   1   1   Rebal. Surface 1: I-faces
  REBZ   1   1   5   1  10   1   1   Rebal. Surface 1: K-faces
  REBM   2   6   9   1  10   1   1   Rebal. Region 2
  REBM   2   1   7   1  10   2   2   Rebal. Region 2
  REBX   2   7   7   1  10   2   2   Rebal. Surface 2: I-faces
  REBZ   2   8   9   1  10   1   1   Rebal. Surface 2: K-faces
  REBZ   2   1   7   1  10   2   2   Rebal. Surface 2: K-faces
  RERM   3   8   9   1  10   2   2   Rebal. Region 3
  REBM   3   1   9   1  10   3   3   Rebal. Region 3
  REBZ   3   1   9   1  10   3   3   Rebal. Surface 3: k-faces
  .....
  .....
  .....
  END

```

Terminate Input Block

Note:

An elipsis (...) indicates that additional user input is required.

must be consistent with the information contained in the Rebalancing Region Location input block.

Mass rebalancing is performed before the first inner iteration during the solution of the pressure equation system. Mass rebalancing is additionally performed before other inner iterations as specified using input variable IREBIT. As shown in Table 11.2, the value input is in terms of a frequency. By way of example, inputting IREBIT=5 specifies that rebalancing is performed before every 5th inner iteration. The frequency to be input is case dependent; it is generally not efficient to perform rebalancing before every inner iteration. A new value may

be specified for IREBIT during a continuation run. The value input for IREBIT must be nonzero.

11.2 Plane-by-Plane Rebalancing

Plane-by-plane rebalancing is similar to the user-specified rebalancing described in Sec. 11.1 except that the code defines the rebalancing regions and each rebalancing region is a plane of fluid cells. The user specifies a rebalancing direction: x (or r), y (or θ), or z ; this determines which type of cell planes are grouped to form rebalancing regions.

The input associated with plane-by-plane rebalancing is in NAMELIST DATA and is shown in Table 11.5. Plane-by-plane rebalancing is selected by inputting a value of 1 for one (and only one) of the three variables IXREB, IYREB, or IZREB; there is no plane-by-plane rebalancing if all three of these variables have a value of 0. Of these three variables, the one variable having the value of 1 indicates the rebalancing direction. For example, inputting IXREB=0, IYREB=0, and IZREB=1 selects plane-by-plane rebalancing in the z direction; fluid cells having spatial location indices of the form (I,J,NR) , where $I=1,2,\dots,IMAX$ and $J=1,2,\dots,JMAX$ (i.e. all cells in the $K=NR$ plane), are in rebalancing region number NR ; this grouping of cells is repeated for $NR=1,2,\dots,KMAX$ to form the rebalancing regions; rebalancing surface NR is composed of the cell faces which separate cells at spatial locations (I,J,NR) from those at spatial locations $(I,J,NR+1)$. The direction of rebalancing should typically be chosen to coincide with the predominant direction of fluid flow through the geometry. Plane-by-plane rebalancing may be selected and deselected at any point in the sequence of runs (i.e. at any value of ISTATE).

Mass rebalancing is performed before the first inner iteration during the solution of the pressure equation system. Mass rebalancing is additionally performed before other inner iterations as specified using input variable IREBIT. As shown in Table 11.5, the value input is in terms of a frequency. By way of example, inputting IREBIT=5 specifies that rebalancing is performed before every 5th inner iteration. The frequency to be input is case dependent; it is generally not efficient to perform rebalancing before every inner iteration. A new value may

Table 11.5 Plane-by-Plane Rebalancing Variables in NAMELIST DATA

Variable	Meaning/Value	Restrictions
IXREB	<p>indicate use of plane-by-plane rebalancing in the x (r) direction; may be:</p> <p>0: not used;</p> <p>1: used.</p>	IYREB=IZREB=0
IYREB	<p>indicates use of plane-by-plane rebalancing in the y (θ) direction; may be:</p> <p>0: not used;</p> <p>1: used.</p>	IXREB=IZREB=0
IZREB	<p>indicate use of plane-by-plane rebalancing in the z direction; may be:</p> <p>0: not used;</p> <p>1: used.</p>	IXREB=IYREB=0
IREBIT	frequency at which rebalancing is performed, i.e. before every IREBIT-th inner iteration.	IREBIT > 0

be specified for IREBIT during a continuation run. The value input for IREBIT must be nonzero.

12. AUXILIARY INPUT

This section discusses the input variables associated with the following auxiliary aspects of running a case: starting, stopping, and restarting a case (Sec. 12.1), selection of solution methods (Sec. 12.2), time step size selection (Sec. 12.3), iteration control (Sec. 12.4), achievement of steady state (12.5), and normal and debug output (Sec. 12.6).

12.1 Starting, Stopping, and Restarting

The input variables associated with starting, stopping, and restarting a case are shown in Table 12.1. The overall indicator of the "state" of the calculation is input using variable ISTATE in NAMELIST DATA. A value of ISTATE=0 indicates the beginning of a steady-state calculation; all input data describing the case must be present in the card-image input file. A value of ISTATE>0 indicates some type of continuation of a previous run: ISTATE=1 continues a steady-state calculation, ISTATE=2 begins a transient calculation from a steady-state calculation, and ISTATE=3 continues a transient calculation. For these continuation runs, the majority of the case information is provided to the code via the restart file; only minimal changes may be made using the card-image input file. For a given steady-state calculation, there will only be one run having ISTATE=0 but any number of runs (including zero) having ISTATE=1; similarly, for a given transient run, there will only be one run having ISTATE=2 but any number of runs (including zero) having ISTATE=3.

Closely related to ISTATE is input variable IFRES in NAMELIST GEOM which describes how restart files are to be used in the current run. A value of IFRES=0 indicates that restart files are not used; this is a permissible (though not necessarily desirable) choice at ISTATE=0. When IFRES has a value of 2, 3, or 4, a restart file is written at the "normal end" of the computer run; the various conditions which define the "normal end" of a computer run are discussed later in this subsection. When IFRES has a value of 1, 3, or 4, a restart file must be provided as input and is read during the initialization stage of the calculation. When IFRES is 1 or 3, all calculational quantities are initialized to the values found on the restart file; when IFRES is 4, only the velocities, fluid temperatures, and pressures are initialized from the restart

Table 12.1 Input Variables Associated with Starting, Stopping, and Restarting a Case in NAMELISTs GEOM and DATA

NAMELIST GEOM:		
Variable	Meaning/Value	Restrictions
IFRES	<p>indicate use of restart files; may be:</p> <p>0: restart files not used;</p> <p>1: write restart file at end;</p> <p>2: read restart file at beginning;</p> <p>3: combines IRES=2 and IFRES=1;</p> <p>4: read velocity components, fluid temperatures, and pressures from restart file as initial guess; write restart file at end.</p>	<p>ISTATE=0</p> <p>ISTATE>0</p> <p>ISTATE>0</p> <p>ISTATE=0 See text.</p>
NAMELIST DATA:		
Variable	Meaning/Value	Restrictions
ISTATE	<p>indicate state of calculation; may be:</p> <p>0: begin steady-state;</p> <p>1: continue steady-state;</p> <p>2: begin transient;</p> <p>3: continue transient.</p>	
TSTART	initial case time [s].	ISTATE=0 or 2
TIMAX	final case time [s].	
NTMAX	final time step number.	
TREST	CPU time [s] to be allowed at end of run for writing restart, etc.	
IBOIL	<p>indicates whether check is made for boiling; may be:</p> <p>0: no check;</p> <p>1: terminate case if $P < P_{\text{sat}}$ in any cell containing Fluid #1.</p>	IFPROP=0

file; this choice is only selectable in conjunction with ISTATE=0; see Sec. 4.2.1 for other restrictions on using the restart file in this manner as a guess. A normal sequence of computer runs will have IFRES=1 in the first run (ISTATE=0) and IFRES=3 in all subsequent runs (ISTATE>0).

The time in the case at the beginning of this computer run is specified using variable TSTART in NAMELIST DATA. This should normally be specified as 0.0 at the beginning of both a steady-state (ISTATE=0) and a transient (ISTATE=2) calculation. A value should not be specified in the card-image input file for runs which continue a calculation in the same mode (ISTATE=1 or ISTATE=3).

The "normal end" of a computer run is achieved by satisfying any one of the several conditions shown in Table 12.2. These include completion of a certain amount of case history (measured in time or time steps), running out of central-processor-unit (CPU) time, initiation of boiling, expansion cell porosity too low, and achievement of steady state. The user inputs for these end conditions are described in the following paragraphs (and most of these are shown in Table 12.1).

Table 12.2 Conditions which Define the "Normal End" of a Computer Run

Condition	User Input
Completion of TIMAX seconds of problem simulation	TIMAX
Completion of NTMAX time steps	NTMAX
Central-Processing-Unit (CPU) time remaining is less than TREST seconds.	TREST
Incipient boiling in Fluid #1	IBOIL
Porosity too low in expansion cell.	---
Achievement of steady state	EPS1,EPS2,EPS3

The input variable TIMAX in NAMELIST DATA specifies the maximum problem time [s]; the value specified for TIMAX is relative to the value specified for TSTART (typically 0.0) at the beginning of a steady-state (ISTATE=0) or transient (ISTATE=2) calculation; the size of the final time step will be shortened, if needed, to hit TIMAX exactly. The input variable NTMAX in NAMELIST DATA specifies the maximum number of time steps for a problem; the value specified for NTMAX is relative to a time step number of 1 as set internally by the code at the beginning of a steady-state (ISTATE=0) or transient (ISTATE=2) calculation. Note that both steady-state and transient calculations are performed by taking a number of time steps, each of which has a finite time step size. Also note that TIMAX and NTMAX do not refer to the incremental time and incremental number of time steps for just this computer run (unless ISTATE is 0 or 2) but for the total case history in the present calculational mode (i.e. either steady state or transient).

The code checks the amount of CPU time remaining (i.e. the difference between the CPU time limit specified for the run and the amount of CPU time consumed by the run to this point) after each outer iteration. When this value becomes less than the value specified by the user via input variable TREST in NAMELIST DATA, the present run will undergo a normal termination. The value input for TREST must be somewhat larger (by 10 to 20 s) than the amount of CPU time required to perform one outer iteration; the value to be input depends on how fast the case is running, which is a function of problem size and severity and the computer speed. Sufficient time must be allowed within TREST for other run termination actions (e.g. final output edits, writing a restart file, file dispositions, etc.). If the value specified for TREST is too small and the job reaches its CPU time limit, the run will undergo an abnormal termination on most computer systems; a restart file will not be written and the user may lose some portion of the printed output. Experience has shown that the combination of a judicious choice for CPU time limit and an appropriate choice for TREST are very useful in managing the execution of individual COMMIX runs.

The user may also stop the calculation at incipient boiling by inputting IBOIL=1 in NAMELIST DATA. Incipient boiling is reached when the fluid pressure (P) is equal to (or just below) the local saturation pressure ($P_{\text{sat}}(T)$). (As a reminder, COMMIX-1AR/P does not treat

two-phase fluid states.) The incipient boiling check only looks at cells which contain fluid number 1, and the check can only be made if "detailed" fluid properties are being used (cf. Sec. 9.1). Although a restart file will be written if requested by the user, a run which terminates due to incipient boiling should not generally be continued.

When using the expansion-cell velocity boundary condition (cf. Sec. 4.1.1), the volume porosity of the expansion cell changes during the course of a transient. Since the equations are only valid for positive volume porosity, checks are made at every time step on whether the volume porosity is getting close to zero. If the volume porosity drops below 0.07, warnings are issued by the code but execution continues. If the volume porosity drops below 0.02 (or is projected to drop below 0.02 during the next time step), the code will print an error message and undergo a "normal end" sequence (a final edit and a restart file are written if requested). This run may only be continued if the volume porosity of the expansion cell is increased by the user (using an 'AL' record in the Internal Cell Initialization input block (cf. Sec. 3.5)) in the subsequent run.

Finally, the achievement of steady-state conditions indicates the normal end of a run in the steady-state mode ($ISTATE \leq 1$). The conditions which define steady state are discussed in Sec. 12.5. The steady-state conditions may be the desired end condition for the calculational sequence. More commonly, these conditions will serve as the initial conditions for performing one or more transient calculations in subsequent runs.

When a run reaches its normal end, a number of actions take place. A restart file is written if requested by the user ($IFRES = 1, 3, \text{ or } 4$); the restart file contains the current values of all calculational variables, allowing a subsequent run to continue the calculation at the next outer iteration (which may be the first outer iteration of the next time step). A final long edit is written to the printable output file if requested by the user (see `NTPRNT` in Sec. 12.6). Other files are closed and control is returned to the operating system for other actions specified by the user.

There are, of course, a number of conditions which will cause an "abnormal end" of a computer run. These include input errors, convergence failures, time step getting too small, and errors detected by the operating system. When an error condition is detected by COMMIX, an informative message is printed on the standard output file which indicates the error, probable cause, and suggested user action; these error conditions are tabulated as a part of the Input Guide in Appendix A; depending on the severity, the run will either continue or terminate. The system generated error messages are usually more terse; the root cause may be difficult to diagnose. When a run undergoes abnormal termination, a restart file is not written and the last portion of the printed output may be lost.

12.2 Selection of Solution Methods

The input variables associated with solution method choices are shown in Table 12.3.

The value input for variable ISYMCH in NAMELIST GEOM selects the method to be used in solving the pressure equation system. This system of equations results from substituting the three momentum conservation equations into the mass conservation equation and is a linear equation system to be solved for the pressures. The user may select from one of three solution procedures; as described in Sec. 11.5 of Ref. 1, these solution procedures are all iterative. The preconditioned conjugate gradient method is selected by ISYMCH=5; this is the recommended choice. A successive overrelaxation method is selected by ISYMCH=3 or ISYMCH=4; the mass residual (δ_p) for each cell m at inner iteration i is computed from Eq. 11.16 of Ref. 1 using

$$\delta_{p,m}^i = a_{0,m}^p P_m^{i-1} - \sum_{\ell=1}^6 (a_{\ell,m}^p P_{\ell}^j) - b_{0,m}^p, \quad (12.1)$$

where a and b are arrays of constants and the summation couples the pressure in cell m to the pressures in its six nearest neighbor cells (indexed symbolically by ℓ). Under ISYMCH=3, the superscript j on the right-hand side of the above equation has the value i (i.e. the value for the

**Table 12.3 Input Variables Associated with Solution Methods in
NAMELISTs GEOM and DATA**

NAMELIST GEOM:		
Variable	Meaning/Value	Restrictions
ISYMCH	select solution method for pressure (combined momentum and mass) equation system; may be: 3: successive overrelaxation; 4: Jacobi-type successive overrelaxation; 5: preconditioned conjugate gradient.	
IFITEN	select solution method for fluid energy equation system; must be: 3: successive overrelaxation.	
NAMELIST DATA:		
Variable	Meaning/Value	Restrictions
IFENER	select fluid energy solution option; may be: 0: no fluid energy calculation; 1: normal calculation; 2: porosity-adjusted conduction lengths.	
SETMO	frequency at which momentum/mass equations are solved; may be: <0: skip solution every SETMO -th time step; >0: perform solution every SETMO-th time step.	See below.
SETEN	frequency at which normal fluid energy equations are solved; may be: <0: skip solution every SETEN -th time step; >0: perform solution every SETEN-th time step.	IFENER > 0 See below.
Restrictions on SETMO and SETEN: (a) SETMO and SETEN must both be nonzero; (b) SETMO=1: SETEN may be any nonzero integer; (c) SETEN=1: SETMO may be any nonzero integer; (d) SETMO<0: SETEN>0 and SETMO must be an integral multiple of SETEN; and (e) SETEN<0: SETMO>0 and SETEN must be an integral multiple of SETMO.		
NTJUMP	frequency with which the turbulence equations are solved, i.e. every NTJUMP-th time step.	ITURKE ≥ 10 NTJUMP > 0

**Table 12.3 Input Variables Associated with Solution Methods in
NAMELISTs GEOM and DATA (Cont'd)**

NAMELIST DATA:		
Variable	Meaning/Value	Restrictions
IREBIT	frequency with which rebalancing is performed, i.e. before every IREBIT-th inner iteration.	IFREB > 0, IXREB > 0, IYREB > 0, or IZREB > 0 and ISYMCH < 5: IREBIT > 0
IHTWAL(99)	coupling of thermal structure and fluid energy equations.	0,1,2 (See Sec. 5.4)

current inner iteration) if the pressure for cell ℓ has already been updated during this inner iteration; otherwise, superscript j has the value $i-1$ (i.e. the value from the previous inner iteration). Under ISYMCH=4, the superscript j always has the value $i-1$ (i.e. the value for the previous inner iteration); this is referred to as a Jacobi-type iteration.

The value input for variable IFITEN in NAMELIST GEOM selects the method to be used in solving the fluid energy equation system. At the present time, there is no choice; the successive overrelaxation method must be used (IFITEN=3). (This method is also used to solve the turbulent kinetic energy (if ITURKE $\geq$ 11) and the turbulent kinetic energy dissipation rate (if ITURKE=12) equation systems.)

Additional input variables associated with the inner iteration methods are described in Sec. 12.4. The remainder of this subsection describes several ways of controlling the frequency with which various sets of equations are solved.

The fluid energy calculation is controled by the input variable IFENER in NAMELIST DATA. The fluid energy calculation is turned off entirely during the run by IFENER=0; under this option, the enthalpy, temperature, and density of each fluid cell remain constant at their initial values throughout the run; this option is often useful in isothermal calculations. The more

typical choice is to perform the fluid energy calculation, selected by $IFENER > 0$. When $IFENER = 1$, the length used for heat conduction in the fluid energy equation is the geometric distance between the centers of adjacent fluid cells. When $IFENER = 2$, the conduction length is the geometric distance times the volume porosity (i.e. each occurrence of Δx , Δy , and Δz in Table 4.3 of Ref. 12.1 is replaced with $\gamma_v \Delta x$, $\gamma_v \Delta y$, and $\gamma_v \Delta z$, respectively).

The frequency at which the momentum/mass and energy equation systems are solved during a run are controlled by the values of input variables $ISETMO$ and $ISETEN$, respectively, in NAMELIST DATA. For each of these input variables, a positive value indicates that the corresponding calculation is to be performed only during time step numbers which are an integral multiple of the input value and a negative value indicates that the corresponding calculation is to be skipped at time step numbers which are an integral multiple of the input value. When a particular system of equations is not solved at a particular time step, the values of certain variables retain their values from the previous time step; for the momentum/mass system, these variables are the three velocity components and the pressure; for the energy system, these variables are the fluid enthalpy, temperature, and density. The normal procedure is to solve both equation systems at every time step (i.e. $ISETMO = 1$ and $ISETEN = 1$). The option of solving one of these two systems less frequently than the other system is occasionally useful when one of the two sets of variables is changing much more slowly than the other during a calculation. When deciding to solve one system less frequently than the other, bear in mind that at least one of the two systems must be solved during each time step; additionally, the user must input $IFENER > 0$.

Two examples of controlling the calculational frequency are shown in Table 12.4. In the first example, the energy calculation is performed every time step ($ISETEN = 1$) and the momentum/mass calculation is only performed every 5th step ($ISETMO = 5$); therefore, the energy system's variables are advanced during time steps 1 through 4 and both the energy and momentum/mass systems' variables are advanced during time step 5; this 5-step pattern repeats during each succeeding block of 5 time steps. In the second example, the energy calculation is skipped every 6th time step ($ISETEN = -6$) and the momentum/mass calculation is performed every 3rd time step ($ISETMO = 3$); therefore, the energy system's variables are advanced during

time steps 1 and 2, both the energy and momentum/mass systems' variables are advanced during time step 3, the energy system's variables are advanced during time steps 4 and 5, and only the momentum/mass system's variables are advanced during time step 6; this 6-step pattern repeats during each succeeding block of 6 time steps.

Table 12.4 Examples of Calculational Frequencies Selected by ISETMO and ISETEN

ISETMO	ISETEN	Equation System(s) Solved ¹ for a Sequence of Time Steps
		11111111... 123456789012345678...
5	1	EEEEEEEEEEEEEB.....
3	-6	EEBEEMEEBEEMEEBEEM...

Note:

1. M=momentum/mass(only), E=energy(only), and B=both(energy and momentum/mass).

A similar type of frequency control is provided for two other parts of the calculation: turbulence and rebalancing.

The value input for NTJUMP in NAMELIST DATA indicates the frequency at which the fluid turbulence equations are solved. When these equations are not solved during a particular time step, one or more of the following variables retain their values from the previous time step depending on which turbulence model is being used (as selected by ITURKE, cf. Sec. 8.1): turbulent viscosity ($ITURKE \geq 10$), turbulent thermal conductivity ($ITURKE \geq 10$), turbulent kinetic energy ($ITURKE \geq 11$), and turbulent kinetic energy dissipation rate ($ITURKE \geq 11$). The normal procedure is to solve the turbulence equations at every time step (i.e. $NTJUMP=1$) when $ITURKE \geq 10$.

The value input for input variable IREBIT in NAMELIST DATA indicates the frequency at which mass rebalancing is performed. In contrast to the other frequencies described above, the frequency specified for mass rebalancing (a) refers to inner iteration numbers rather than time step numbers and (b) is normally a number greater than 1. The value input for this frequency has no effect unless the user has opted to use one of the mass rebalancing schemes, which are described in Sec. 11.

As a reminder, the user has a choice for the type of numeric coupling between the thermal structure and fluid energy equations. This choice is made using input array location IHTWAL(99) and is described in Sec. 5.4.

12.3 Time Step Size Selection

The input variables associated with time step size selection are shown in Table 12.5. All of these variables are in NAMELIST DATA. The value input for variable IDTIME selects the manner in which the time step size is adjusted. There are three choices: manual (IDTIME=0), Courant-based automatic (IDTIME=1), and changed-based automatic (IDTIME=2), which are described in Sec. 11.4 of Ref. 1.

Under the manual mode (IDTIME=0), the user supplies two time step sizes, Δt_1 and Δt_2 , using input variables DT(1) and DT(2), respectively, and a time step number for switching, N_{t0} , using input variable LASTDT. The first of these two time step sizes (Δt_1) is used for all time steps through time step number N_{t0} ; the second of these two time step sizes (Δt_2) is used for time step numbers after N_{t0} .

Under the Courant-based automatic mode (IDTIME=1), the time step size is adjusted based on the Courant condition, which relates the time step size to the number of mesh cells a mass particle would traverse in one time step. The time step size is set using

$$\Delta t = \frac{\beta_1}{\left(\frac{A_i u_i}{V_f} \right)_{\max}}, \quad (12.2)$$

Table 12.5 Input Variables Associated with Time Step Size Selection in NAMELIST DATA

Variable	Meaning/Value	Restrictions
IDTIME	select time step size control method; may be: 0: manual: time step size is DT(1) through step LASTDT, then DT(2); 1: Courant-based automatic: time step size is set internally to the maximum allowed by Courant limit multiplied by RDTIME; 2: change-based automatic: time step size is adjusted between limits DT(3) and DT(4) to keep maximum step-to-step changes in key quantities between CHGMIN and CHGMAX (see text).	
DT(1)	Δt_1 : initial time step size [s].	
DT(2)	Δt_2 : time step size [s] to use after step LASTDT. (Do not input at ISTATE=1 or 3 if IDTIME=2.)	IDTIME $\neq$ 1
DT(3)	$\Delta t_{\min}$: minimum time step size [s].	IDTIME=2 DT(3) < DT(4)
DT(4)	$\Delta t_{\max}$: maximum time step size [s].	IDTIME=2 DT(3) < DT(4)
LASTDT	N_0 : time step after which to switch time step size from DT(1) to DT(2).	IDTIME=0
RDTIME	β_1 : factor [-] which multiplies Courant-based maximum time step size to get actual time step size.	IDTIME=1
CHGMIN	$\epsilon_{i,\min}$: if maximum fractional change per step is less than CHGMIN for N_2 consecutive steps, increase time step size by DTFAC. (Value of N_2 is set internally by code; initial value is 5.)	IDTIME=2 CHGMIN < CHGMAX
CHGMAX	$\epsilon_{i,\max}$: if maximum fractional change per step is greater than CHGMAX, cut time step size by DTFAC and repeat time step.	IDTIME=2 CHGMIN < CHGMAX
DTFACT	β_2 : factor [-] by which to increase or decrease time step size.	IDTIME=2 DFTACT > 1

where β_1 is a constant input using variable RDTIME. The quantity $(A_i u_i / V_f)_{\max}$ is the maximum volumetric flow rate divided by the cell fluid volume; the maximum value of this quantity is found by sweeping over all six faces of all fluid cells in the computational grid. The

time step size is adjusted using this formula at the time step numbers provided by the user in input array NTHCON, which is discussed further in Sec. 12.4. This mode of time step selection had a strong theoretical basis when the numerics in COMMIX were explicit in time, which would dictate $\beta_1 < 1$ and, thus, would restrict a particle of mass from completely traversing a cell during a time step. There is less theoretical basis for choosing β_1 for the current implicit-in-time formulation, although arguments can be made for $\beta_1 \gg 1$.

Under the change-based automatic mode (IDTIME=2), the time step size is adjusted based on the relative rate of change of certain variables. Specifically, the step-to-step changes of the kinetic energy (E), fluid temperature (T), and turbulent viscosity (μ_t) are evaluated using the following relational expressions:

$$\varepsilon_{t,\min} < \frac{|E^n - E^{n-1}|}{\max(E^n)} < \varepsilon_{t,\max} , \quad (12.3)$$

$$\varepsilon_{t,\min} < \frac{|(T^n + 273.15) - (T^{n-1} + 273.15)|}{(T^n + 273.15)} < \varepsilon_{t,\max} , \quad (12.4)$$

and

$$\varepsilon_{t,\min} < \frac{|(\mu_t + \mu_t)^n - (\mu_t + \mu_t)^{n-1}|}{(\mu_t + \mu_t)^n} < \varepsilon_{t,\max} , \quad (12.5)$$

where superscripts n and n-1 denote the current and previous time step numbers, respectively. The kinetic energy of a fluid cell is used as an aggregate measure of fluid velocity and is defined by

$$E \equiv \rho V_f (\bar{u}^2 + \bar{v}^2 + \bar{w}^2) , \quad (12.6)$$

where $\bar{u}$, $\bar{v}$, $\bar{w}$, are center-of-mass-cell averages of the velocity components in the three spatial directions. The maximum cell kinetic energy, found by sweeping over all fluid cells, is used for normalization purposes in Eq. 12.3 so that fluctuations of very small velocities are ignored

for the purpose of adjusting the time step size. Turbulent viscosity is a measure of both the turbulent kinetic energy (k) and the turbulent kinetic energy dissipation rate (ϵ); the total viscosity is used in Eq. 12.5 since this is the main term which provides coupling between the normal conservation equations and the turbulence transport equations; if the turbulent viscosity is small relative to the laminar viscosity, then Eq. 12.5 will not be a controlling factor in time step size selection; if one of the constant turbulent viscosity models (i.e. ITURKE < 10, cf. Sec. 8.1) is being used, then Eq. 12.5 is not used for time step size selection. The quantities $\epsilon_{t,min}$ and $\epsilon_{t,max}$ are input by the user using variables CHGMIN and CHGMAX, respectively.

Under the change-based automatic mode, the relational expressions given by Eqs. 12.3 through 12.5 are evaluated after the outer iteration has converged (i.e. at the end of the time step). If the relational expressions are all satisfied, then the time step size remains unchanged for the next time step. If all of the changes are less than $\epsilon_{t,min}$ for N_2 consecutive time steps, then the time step size to be used for the next time step will be increased using

$$\Delta t^{n+1} = \min (\beta_2 \Delta t^n, \Delta t_{max}) , \quad (12.7)$$

where β_2 is a constant input using variable DTFACK and Δt_{max} is a maximum time step size input using variable DT(4); the quantity N_2 is set by the code and initially has a value of 5. If any of the changes exceeds $\epsilon_{t,max}$, then the time step size is reduced using

$$\Delta t = \frac{\Delta t^n}{\beta_2} , \quad (12.8)$$

where β_2 is the same constant used in Eq. 12.7. If the resulting time step size is greater than the minimum time step size (Δt_{min}) input using variable DT(3), all variables are restored to their beginning of time step values and the current time step is repeated; otherwise, the calculation is terminated with an error message. During a steady-state calculation, the logic controlling time step size decrease is bypassed in the code until after the time step size has undergone at least one increase (i.e. the changes are less than $\epsilon_{t,min}$ for N_2 consecutive steps); this avoids needlessly decreasing the time step size early in the calculation where quantities are expected to be changing by rather large amounts; during this early stage of the calculation, the time step size is set by the code using what was described above for the manual mode (i.e. switching from Δt_i

to Δt_2 at step number N_{10}). During a transient, the step repeat path is also taken if the outer iteration does not converge in the allowed maximum number of iterations (cf. Sec. 12.4). If, after any time step size increase, the code during the next time step finds that the time step must be repeated using a reduced time step size, then the code will double the current value of N_2 ; this will slow down the rate at which the time step size can be increased and avoid some of the step repeats encountered in ill-behaved cases.

The user will normally want to select the change-based mode (i.e. IDTIME=2). Steady-state calculations usually need to be started using small time step sizes (e.g. $\Delta t \leq 0.1$ s). During a transient calculation, the initial time step size and the limits on time step size depend on the severity of the transient (e.g. how fast user-input time-dependent boundary conditions are changing).

12.4 Iteration Controls and Parameters

Advancement of variables across a time step requires a sequence of one or more outer iterations. Each outer iteration contains up to four different inner iteration processes. The various parameters associated with these inner and outer iteration processes are input in NAMELIST DATA and are shown in Table 12.6.

Inner iteration processes are used to solve four equation systems: pressure (i.e. combined momentum and mass), turbulent kinetic energy, turbulent kinetic energy dissipation rate, and fluid energy. Previously described input variables (i.e. IFENER, ISETMO, ISETEN, ITURKE, and NTJUMP) determine which of these iterative processes are being performed and which are being skipped during a run or during a specific time step within a run. With the exception of the pressure equation system, these inner iterations are all performed using a successive overrelaxation technique; the pressure equation system may be solved using either successive overrelaxation or conjugate gradient (cf. ISYMCH in Sec. 12.2). Each inner iteration process continues until a user-specified degree of convergence is obtained or until a user-specified maximum number of iterations is reached.

Table 12.6 Input Variables Associated with Iteration Control in NAMELIST DATA

Variable	Meaning/Value	Restrictions
NTHCON(N)	array of time step numbers at which to call SUBROUTINE GDCONV, which calculates (a) time step size under IDTIME=1 and (b) convergence criteria for pressure equation system; entries may be: <0 : call GDCONV every NTHCON th time step; no further values in array are processed; $=0$: no further calls to GDCONV; >0 : call GDCONV at indicated step; after this step, use the next entry in the array.	$N=1,2,\dots,10$
EPS1	ϵ_1 : multiplicative coefficient in inner iteration convergence criterion for pressure equation [-].	$0 < EPS1 \leq 1$ See text.
EPS2	ϵ_2 : additive term (mask) in inner iteration convergence criterion for pressure equation [$\text{kg/m}^2\text{s}$].	$0 < EPS2$ See text.
EPS5	ϵ_h : inner iteration convergence criterion for fluid enthalpy [-].	$0 < EPS5 \leq 1$
EPS6	ϵ_k and ϵ_ϵ : inner iteration convergence criterion for turbulent kinetic energy and turbulent kinetic energy dissipation rate [-].	$0 < EPS6 \leq 1$
ITMAXE	maximum number of inner iterations (per outer iteration) allowed to solve energy equation system.	
ITMAXK	maximum number of inner iterations (per outer iteration) allowed to solve turbulent kinetic energy and turbulent kinetic energy dissipation rate equation systems.	ITURKE > 10
ITMAXP	maximum number of inner iterations (per outer iteration) allowed to solve pressure equation system.	
OMEGA	ω : relaxation factor for solution of the pressure equations [-].	$0 < OMEGA < 2$ ISYMCH=3
RELAXE	ω_h : relaxation factor for solution of the energy equations [-].	$0 < RELAXE \leq 1$
RELAXK	ω_k and ω_ϵ : relaxation factor for solution of the turbulent kinetic energy and the turbulent kinetic energy dissipation rate transport equations [-].	$0 < RELAXK \leq 1$
OMEGAD	Ω_ϵ : relaxation factor for coefficients in the turbulent kinetic energy transport equation [-].	
OMEGAE	Ω_h : relaxation factor for coefficients in the energy transport equations [-].	
OMEGAK	Ω_k : relaxation factor for coefficients in the turbulent kinetic energy transport equation [-].	
OMEGAV	Ω_v : relaxation factor for coefficients in the momentum transport equations [-].	

**Table 12.6 Input Variables Associated with Iteration Control
in NAMELIST DATA (Cont'd)**

Variable	Meaning/Value	Restrictions
EPS3	ϵ_3 : outer iteration convergence criterion for fluid velocity components and enthalpy [-].	$0 < \text{EPS3} \leq 1$
IT(1)	maximum number of outer iterations per time step through time step LASTIT.	
IT(2)	maximum number of outer iterations per time step after time step LASTIT.	
LASTIT	time step number after which to switch from IT(1) to IT(2) outer iterations per time step.	
OMR	Ω_r : unerelaxation factor for advancement of radiation heat transfer rate across a time step [-].	$0 < \text{OMR} \leq 1$
OMEGAT	Ω_t : underrelaxation factor for advancement of turbulent viscosity across a time step [-].	$0 < \text{OMEGAT} \leq 1$

The convergence criterion for the pressure equation system is specified in terms of the value below which the mass residual ($\delta_{p,m}$) must be reduced for each fluid cell m ; symbolically, this criterion is expressed as

$$\frac{\delta_{p,m}^i}{V_f} < \left[\frac{\delta}{V} \right]_{p,ref}, \quad (12.9)$$

where

$$\left[\frac{\delta}{V} \right]_{p,ref} \equiv \epsilon_1 \left[\left[\frac{\rho A_i u_i}{V_f} \right]_{\max} + \epsilon_2 \right], \quad (12.10)$$

where superscript i in Eq. 12.9 denotes inner iteration i . The quantity $(\rho A_i u_i / V_f)_{\max}$ in Eq. 12.10 is the maximum mass flow rate per cell fluid volume; the maximum value of this quantity is found by sweeping over all six faces (subscript i) of all fluid cells in the computational grid.

The quantities ε_1 and ε_2 are specified by the user using input variables EPS1 and EPS2, respectively; note that ε_2 has units of $[\text{kg}/\text{m}^3 \cdot \text{s}]$ and may be used as a bias or masking factor. The user will normally want to require rather tight convergence of the pressure equation system by specifying ε_1 to be small (e.g. 10^{-9}) and ε_2 to be much smaller than $(\rho A_i u_i / V_t)_{\max}$.

A new value for $(\delta/V)_{p,\text{ref}}$ is computed at the time step numbers input by the user in array NTHCON. A positive value in NTHCON indicates a time step number at which to evaluate Eq. 12.10; after this time step number, the next entry in the array will be used. A negative value in NTHCON indicates a frequency at which Eq. 12.10 is to be evaluated; values following a negative entry are not processed by the code. Up to 10 values may be entered in array NTHCON; the final entry made by the user in this array should be 0 if less than 10 values are entered. By way of example, if the user inputs

$$\text{NTHCON} = 1, 25, -100, 2000, 0,$$

then Eq. 12.10 will be evaluated at time step numbers 1 and 25 and then every 100th time step (i.e. time step numbers 100, 200, 300, ...); the final two entries in the array in this example (2000, 0) are never used since they are preceded by a negative entry. The number and values of entries input by the user in array NTHCON depend on the case; Eq. 12.10 must be reevaluated most frequently when $(\rho A_i u_i / V_t)_{\max}$ is changing most rapidly for the resulting value of $(\delta/V)_{p,\text{ref}}$ to be used meaningfully as a convergence criterion.

For the other inner iteration processes, convergence is obtained when the change in the unknowns between successive inner iterations is less than a specified value; symbolically, this is expressed by

$$\frac{\phi_m^i - \phi_m^{i-1}}{\phi_m^i} < \varepsilon_\phi \quad (12.11)$$

For the fluid energy equation system ($\phi=h$), ε_h is input using variable EPS5. For the turbulent kinetic energy ($\phi=k$) and turbulent kinetic dissipation rate ($\phi=\epsilon$) equation systems, the value

of input variable EPS6 is used to set both ε_k and ε_ε . The user will generally want to require rather tight convergence of these inner iteration processes (e.g. $\varepsilon_\phi = 10^{-9}$).

The user also specifies the maximum number of iterations allowed to obtain convergence. These are specified using input variable ITMAXP for the pressure equation system, ITMAXE for the fluid energy equation system, and ITMAXK for the turbulent kinetic energy and turbulent kinetic dissipation rate equation systems. If the maximum number of iterations is reached without convergence being obtained, a warning message is printed but the run is not terminated. The user should allow sufficient inner iterations for the specified degree of convergence to actually be obtained. The number of inner iterations needed to obtain convergence is a function of the number of fluid cells, the convergence criteria, and the severity of the transient and, thus, may range from only a few up to several hundred.

In the successive overrelaxation method, a sweep is made through all fluid cells (m) to calculate the residual (δ) of the conservation equation for variable ϕ at inner iteration i from

$$\delta_{\phi,m}^i = a_{0,m}^\phi \phi_m^{i-1} - \sum_{l=1}^L (a_{l,m}^\phi \phi_l^j) - b_{0,m}^\phi, \quad (12.12)$$

where ϕ stands for P, k, ε , or h and a and b are arrays of known coefficients. The superscript j is either i or i-1 as described in connection with Eq. 12.1. The residual is then used to compute a new value for the solution variable for cell m from

$$\phi_m^i = \phi_m^{i-1} - \frac{\omega_\phi \delta_{\phi,m}^i}{a_{0,m}^\phi}, \quad (12.13)$$

where ω_ϕ is a relaxation constant. The relaxation constants are specified by the user for each equation system using variable OMEGA for ω_p in the pressure equation system, RELAXE for ω_h in the fluid energy equation system, and RELAXK for ω_k and ω_ε in the turbulent kinetic energy and turbulent kinetic energy dissipation rate equation systems. What has been identified here as "successive overrelaxation" is really only overrelaxed if $\omega > 1$; the method is underrelaxed if $\omega < 1$.

The residuals of the conservation equations are nominally defined by Eq. 12.12. The user is given the option of altering the relative importance of the diagonal terms in these equation systems through constants Ω_ϕ , such that the residual is computed from

$$\delta_{\phi,m}^i = \left[\frac{1}{\Omega_\phi} \right] a_{0,m}^\phi \phi_m^{i-1} - \sum_{t=1}^L (a_{t,m}^\phi \phi_t^i) - \left[b_{0,m}^\phi + \left[\frac{1-\Omega_\phi}{\Omega_\phi} \right] a_{0,m}^\phi \phi_m^0 \right], \quad (12.14)$$

where ϕ_m^0 is the value of the unknown for cell m before the inner iteration process begins and remains constant during the inner iteration process. If $\Omega_\phi=1$, the expression given by Eq. 12.14 is identical to that given by Eq. 12.12. The values of Ω_ϕ are supplied by the user using input variable OMEGAE for Ω_h in the fluid energy equation system, OMEGAK for Ω_k in the turbulent kinetic energy equation system, and OMEGAD for Ω_ϵ in the turbulent kinetic energy dissipation rate equation system. In a similar manner, the rearrangement of the momentum equations to express the unknown velocity component (ϕ , which stands for u , v , or w) as a function of the unknown difference in pressures between the centers of adjacent mass cells (ΔP), which is normally written as (cf. Eq. 5.2 of Ref. 1)

$$\phi = \hat{\phi} - \hat{d}^\phi \Delta P, \quad (12.15)$$

is actually computed using

$$\phi = \Omega_v (\hat{\phi} - \hat{d}^\phi \Delta P) + (1-\Omega_v) \phi, \quad (12.16)$$

where $\hat{\phi}$ and $\hat{d}^\phi$ are arrays of known coefficients. The ϕ on the right-hand side of Eq. 12.16 is the value of the velocity component from the previous outer iteration (i.e. known). The value of Ω_v is input by the user using variable OMEGAV. When $\Omega_v=1$, Eq. 12.16 reduces to Eq. 12.15.

The sequence of calculations performed within the outer iteration process is described in Sec. 11.2 of Ref. 1. As was the case with the inner iteration processes, the outer iteration process continues until a user-specified degree of convergence is obtained or until a user-specified maximum number of iterations is reached; additionally, there are two relaxation constants employed.

Convergence of the outer iteration is obtained when the change in the unknowns between successive outer iterations is less than a specified value; specifically, these convergence tests involve the three velocity components (u , v , and w) and the fluid enthalpy (h). Symbolically, these convergence checks are given by

$$\left| \frac{u_m^{n,o} - u_m^{n,o-1}}{[(\bar{u}^2 + \bar{v}^2 + \bar{w}^2)^{0.5}]_{\max}} \right| < \epsilon_3, \quad (12.17)$$

$$\left| \frac{v_m^{n,o} - v_m^{n,o-1}}{[(\bar{u}^2 + \bar{v}^2 + \bar{w}^2)^{0.5}]_{\max}} \right| < \epsilon_3, \quad (12.18)$$

$$\left| \frac{w_m^{n,o} - w_m^{n,o-1}}{[(\bar{u}^2 + \bar{v}^2 + \bar{w}^2)^{0.5}]_{\max}} \right| < \epsilon_3, \quad (12.19)$$

and

$$\left| \frac{h_m^{n,o} - h_m^{n,o-1}}{h_m^{n,o}} \right| < \epsilon_3, \quad (12.20)$$

where subscript m denotes the fluid cell index, superscript n is the time step number, and superscripts o and $o-1$ denote the current and previous outer iteration index. The quantity $[\dots]_{\max}$ in Eqs. 12.17 through 12.19 is the maximum fluid speed, calculated by sweeping over all fluid cells and computing the center-of-mass-cell spatial average of all three velocity components. The convergence criterion ϵ_3 is specified by the user using input variable EPS3; the value of ϵ_3 should be small (e.g. $< 10^{-5}$) but not smaller than the value specified for the inner iteration convergence criteria. In addition to the the four criteria given by Eqs. 12.17 through 12.20, the mass residuals (δ_p) must all satisfy Eq. 12.9 (implying that the inner iteration process for the pressure equation system converged during this outer iteration) in order for the outer iteration to be considered converged.

The user also specifies the maximum number of outer iterations allowed to obtain convergence using input variables IT(1), IT(2), and LASTIT. Time step numbers 1 through LASTIT are allowed IT(1) outer iterations to converge; time step numbers after LASTIT are allowed IT(2) outer iterations to converge. For a steady-state calculation ($ISTATE \leq 1$), the user should specify IT(1)=1 and IT(2)=1; steady state is achieved by performing multiple time steps, each of which has only one outer iteration per time step. For a transient calculation, the user should allow sufficient outer iterations for convergence to actually be achieved; a nominal input value for IT(1) and IT(2) is 20; however, the number of outer iterations required is a function of the time step size (Δt), the convergence criteria (ϵ_1 , ϵ_2 , and ϵ_3), and the severity of the transient. If convergence is not obtained in the maximum number of outer iterations allowed, a warning message is printed; when using the change-based automatic time step control (IDTIME=2), the time step size will be reduced and the step repeated; under the other time step control options, the calculation will proceed to the next time step. Transient calculations in which the outer iterations are routinely not converging should be discontinued and restarted using a smaller time step size or allowing more outer iterations to achieve the desired degree of convergence.

As a reminder, there are two parts of the calculation where an option is provided to set the value at the new time step in an underrelaxed fashion, which essentially performs a weighted average between the new value from the difference equation and the value from the previous time step. One of these is the radiation heat flux, which is described in Sec. 5.4; the underrelaxation factor (Ω_r) is input using variable OMR in NAMELIST DATA. The other is the turbulent viscosity, which is described in Sec. 8.3; the underrelaxation factor (Ω_v) is input using variable OMEGAT in NAMELIST DATA. The user is referred to the indicated sections of this report for additional details.

12.5 Steady State

When running in the steady-state mode (i.e. $ISTATE \leq 1$), the code solves the same equations in the same order as for a transient calculation. A time step size must be specified (cf. DT and IDTIME in Sec. 12.3); this generally needs to start small but may usually be increased as the equations move toward convergence. The solution is advanced toward steady state somewhat just by virtue of there being multiple time steps; however, a significant problem time (not just number of time steps) must elapse for the solution to converge.

Specified-value-type boundary conditions and heat sources should be constant in time during the steady-state calculation. Any "transient functions" referred to in the input file (cf. Sec. 4.1.4) should be defined as having a constant value (of 1, typically) using entries of the form

TVAL=0.,1.0E20, 0.,1.0E20, etc.,

FVAL=1.,1., 1.,1., etc.,

NEND=2, 2, etc.

This section of input is then easily changed by the user at the beginning of a transient calculation ($ISTATE=2$) to specify the true time dependent functions.

The inner iteration parameters for a steady-state case should be chosen in the same manner as for a transient case; i.e., small ϵ_i and sufficient inner iterations allowed for this convergence to be obtained (cf. Sec. 12.4). On the other hand, the maximum number of outer iterations per time step should be set to 1 in the steady-state mode (i.e., $IT(1)=1$ and $IT(2)=1$); rather than taking multiple outer iterations per time step, steady state is approached by taking multiple time steps with each having only one outer iteration.

During the approach to steady state, the code is continually evaluating the step to step changes in the fluid velocities and enthalpies; symbolically, these tests are given by

$$\left| \frac{u_m^n - u_m^{n-1}}{[(\bar{u}^2 + \bar{v}^2 + \bar{w}^2)^{0.5}]_{\max}} \right| < \epsilon_3 , \quad (12.21)$$

$$\left| \frac{v_m^n - v_m^{n-1}}{[(\bar{u}^2 + \bar{v}^2 + \bar{w}^2)^{0.5}]_{\max}} \right| < \epsilon_3 , \quad (12.22)$$

$$\left| \frac{w_m^n - w_m^{n-1}}{[(\bar{u}^2 + \bar{v}^2 + \bar{w}^2)^{0.5}]_{\max}} \right| < \epsilon_3 , \quad (12.23)$$

and

$$\left| \frac{h_m^n - h_m^{n-1}}{h_m^n} \right| < \epsilon_3 , \quad (12.24)$$

where subscript m denotes the fluid cell index and superscripts n and $n-1$ denote the current and previous time step number. (These tests are identical to those given by Eqs. 12.17 through 12.20 for convergence of the outer iteration process in a transient calculation except that the changes are step-to-step rather than outer-to-outer.) The convergence criterion ϵ_3 is specified by the user using input variable EPS3; the value of ϵ_3 should be small (e.g. $< 10^{-5}$) but not smaller than the value specified for the inner iteration convergence criteria. In addition to the four criteria given by Eqs. 12.21 through 12.24, the mass residuals (δ_p) must all satisfy Eq. 12.9 (implying that the inner iteration process for the pressure equation system converged during this time step) for steady state to be achieved. Once these convergence tests on the velocity components, enthalpy, and mass residual are all satisfied, the code will print a message that "steady state has been achieved" and the run will undergo a "normal end". Since these checks do not cover all calculational variables (e.g. thermal structure temperatures and turbulence quantities are not included), the user should verify from a detailed examination of the code output that steady state has, indeed, been achieved.

12.6 Output Controls

During the course of a run, the code generates a printable output file and, optionally, two binary output files -- restart and plot. The input variables which control the amount of information on the printable output file and the binary plot file are discussed below. The user control over writing the binary restart file (using input variable IFRES) was discussed in Sec. 12.1.

An example of a printable output file (file name OUTPUT on UNIT number 6) is reproduced in Appendix E (Table E.3). There are three categories of information written to this file: case summary, short edits for each time step, and long edits for selected time steps. These categories are discussed below.

The printable output file begins with a case summary. This begins with a header page identifying the code version. This is followed by a listing of the card-image input file. This is followed by various summaries of the geometry, boundary conditions, properties, thermal structures, force structures, pumps, and iteration and time step control information. The amount of detail shown in these summaries depends on ISTATE; more information is shown at the beginning of a steady-state or transient run (ISTATE=0 or 2) than is shown for continuation runs (ISTATE=1 or 3). The user has control over only four parts of these summaries using the input variables shown in Table 12.7; of these, having IBSBUG > 0 is useful for obtaining a quick look at the placement of boundary surfaces; nonzero entries for the other three variable are generally only of use to the code developers during debugging. At the beginning of a steady-state or transient run (ISTATE=0 or 2), the case summary section is followed by a long edit, which is described later in this subsection, showing initial values.

During a calculation, a short edit is written for each time step. At a minimum, this is a single line containing the time step number, the problem time, the time step size, the number of outer iterations taken for this step, the maximum mass residual (normalized by the right-hand side of Eq. 12.9), the maximum value of the changes in the three velocity components and the enthalpy (i.e. the left-hand side of Eqs. 12.21 through 12.24), and the amount of CPU time remaining. Additional information (Courant-based time step size, right-hand side of Eq. 12.9,

**Table 12.7 Input Variables Associated with Case Summary Output
Control in NAMELISTs GEOM and STRUCT**

NAMELIST GEOM:		
Variable	Meaning/Value	Restrictions
IBSBUG	select printing of "Boundary Surface Summary" (a picture showing all boundary locations) at beginning of run; may be: 0: not printed; 1: print and continue execution; 2: print and terminate execution.	
ISTBUG	select printing of "Storage Allocation Table" (shows allocation of variable-dimension arrays within COMMON BLOCK /SPACE/) at beginning of run; may be: 0: not printed; 1: printed.	
LMPRNT	select printing of "Cell Number" and "Surface Number" arrays at beginning of run; may be: 0: not printed (continue execution); 1: print cell number array and terminate execution; 2: print cell and surface number arrays and terminate execution.	
NAMELIST STRUCT:		
Variable	Meaning/Value	Restrictions
ITSBUG	select printing of thermal structure model storage pointers at beginning of run; may be: 0: not printed; 1: printed.	ISTRUC=2

and maximum fluid speed used in the denominator of Eqs. 12.21 through 12.23) is printed on this line for the time steps indicated in array NTHCON (cf. Sec. 12.4). Additional information may be requested by the user for each time step, for each outer iteration, and for each inner iteration. The user input variables that activate these additional prints are shown in Table 12.8.

**Table 12.8 Input Variables Associated with Debug Output Control in
NAMELISTs DATA and PUMP**

NAMELIST GEOM:		
Variable	Meaning/Value	Restrictions ¹
IDLBUG	debug flag: pressure equation.	0,99,IJJKK
IENBUG	debug flag: energy equation.	0,1,2,99,IJJKK
IINBUG	debug flag: hex fuel bundle input.	0,1,2
IMOBUG	debug flag: pressure equation.	0,1,2,3
ITDBUG	debug flag: turbulent kinetic energy dissipation rate equation.	0,1,99,IJJKK
ITIBUG	debug flag: general.	-3,0,1,2,3,4,5
ITKBUG	debug flag: turbulent kinetic energy equation.	0,1,3,99,IJJKK
IXMBUG	debug flag: x(r)-direction momentum equation.	0,99,IJJKK
IYMBUG	debug flag: y(θ)-direction momentum equation.	0,99,IJJKK
IZMBUG	debug flag: z-direction momentum equation.	0,1,99,IJJKK
NAMELIST PUMP:		
Variable	Meaning/Value	Restrictions ¹
IPUBUG	debug flag: pump model.	0,1,2,3,4,5

Note:

1. The allowed values generally conform to the following hierarchy: 0 is no print; 1 is minimal print; 2, 3, etc. is sequentially more print; 99 prints values for all cells; and a six-digit integer of the form IJJKK prints values for the cell at spatial location (II,JJ,KK).

The values input for these variables generally follow a hierarchy that the amount of information printed goes up as the value goes up; this can result in the short edit actually being much longer than the long edit. In general, the user will not want to specify nonzero values for these variables; they are primarily for use in activating prints to be used by the code developers. There are two exceptions: (1) The combination of ITIBUG=2, IMOBUG=1, and IENBUG=1 is useful in providing some information concerning the progression of inner and outer iterations (i.e. the number of iterations actually taken and which cell has the maximum error). (2) The only way to print variables associated with pumps is by having IPUBUG>0. Somewhat more detail on the output generated by these debug flags is provided in the Input Guide (Appendix A); a more detailed discussion is in the Programmer's Guide (Ref. 14).

The user may cause additional long edits to be printed at specific time step numbers or problem times using input arrays NTPRNT and TPRNT, respectively, as shown in Table 12.9. The input for array NTPRNT is similar to that for array NTHCON (cf. Sec. 12.4): a positive entry indicates a specific time step number and a negative entry indicates a frequency. The special entry of '-9999' indicates that a long edit is to be generated after the last time step in the run; this is particularly useful during steady-state runs which stop on CPU time (cf. input variable TREST in Sec. 12.1); however, the final long edit is of little use in a transient run which stops on CPU time before completely converging the time step. Up to 49 values may be entered in array NTPRNT; the final entry made by the user in this array should be 0 if less than 49 values are entered. By way of example, if the user inputs

NTPRNT=5,50,100,-9999,0,

long edits will be generated at the end of time step numbers 5, 50, and 100; if the run continues beyond time step number 100, a long edit will be generated after the final time step (or final outer iteration) in the run; however, if the run stops between time step numbers 50 and 100, the last long edit will be that for time step number 50. The input for array TPRNT is similar to that for array NTPRNT with two major exceptions: (1) a negative entry indicates a frequency that takes effect at the time specified by the next entry in the array and (2) the entries are floating-point numbers rather than integers. The code does not adjust the time step size in order to hit a requested print time exactly; therefore, long edits will be generated at the end of the time steps on which the problem times equal or first exceed the times requested in array TPRNT. Up to 50 values may be entered in array TPRNT; the final entry made by the user in this array should be 0.0 if less than 50 values are entered. By way of example, if the user inputs

TPRNT=1.,5.,-25.,100.,0.,

full edits will be generated at (or just after) problem times of 1 s, 5 s, 100 s, and every 25 s thereafter (i.e. 125 s, 150 s, 175 s, etc.). Nonzero entries in both the NTPRNT and TPRNT arrays may be used to combine the features of printing by time step number and by problem time.

Long edits are generated at the beginning of a steady-state or transient calculation (ISTATE=0 or 2) and at other time steps or problem times during a calculation as requested by the user using input arrays NTPRNT and TPRNT. A long edit begins with several summaries of calculated values: boundary surfaces, thermal structures, radiation surfaces, and global mass and energy balances; the user has no control over this part of a long edit. The additional information to be shown in a long edit is controlled by the values specified in array ISTPR for

Table 12.9 Input Variables Associated with Normal Output Control in NAMELIST DATA

Variable	Meaning/Value	Restrictions
NTPRNT(N)	<p>array of time step numbers at which information is written to the Print file (filename=stdout, unit=6); entries may be:</p> <p>= -9999: write just before termination of this run; no further values in array are processed;</p> <p>< 0: write at every NTPRNT th time step; no further values in array are processed;</p> <p>= 0: no further writes;</p> <p>> 0: write at indicated step; after this step, use the next entry in the array.</p>	<p>N = 1, 2, ..., 49</p> <p>NTPRNT $\neq$ -9999</p>
TPRNT(N)	<p>array of problem times [s] at which information is written to the Print file (filename=stdout, unit=6); entries may be:</p> <p>< 0: write at every TPRNT th time interval beginning after (or at) the time specified by the next entry in the array; no further values in array are processed;</p> <p>= 0: no further writes;</p> <p>> 0: write at (or after) indicated problem time; after this time, use the next entry in the array.</p>	N = 1, 2, ..., 50
The frequency of printing is controlled by the combination of NTPRNT and TPRNT. The amount of information printed each time is determined by ISTPR and NTHPR.		
ISTPR(N)	array of values indicating which portions of which arrays are to be displayed before taking the first time step; allowed values are described under NTHPR.	<p>N = 1, 2, ..., 50</p> <p>ISTATE = 0 or 2</p>

**Table 12.9 Input Variables Associated with Normal Output Control in
NAMELIST DATA (Cont'd)**

Variable	Meaning/Value	Restrictions
NTHPR(N)	<p>array of values indicating which portions of which arrays are to be displayed at the time steps and/or problem times indicated in arrays NTPRNT and TPRNT; allowed values are signed 5-digit integers of the form:</p> <p>±VVPLL: internal array VV for plane LL of type P,</p> <p>±9VVLL: surface array VV for boundary surface LL,</p> <p>±8NNNN: thermal structure prototype NNNN,</p> <p>or</p> <p>0: no more values in array are to be processed,</p> <p>where</p> <p>+ : only print indicated plane, surface, or thermal structure;</p> <p>- : print all planes, surfaces, or thermal structures between the one indicated by this value and the one indicated by the next value in this array;</p> <p>VV : array identifier;</p> <p>P : type of plane; may be:</p> <p>1: I-plane(s);</p> <p>2: J-plane(s);</p> <p>3: K-plane(s);</p> <p>LL : plane or boundary surface index number;</p> <p>NNNN : structure to be printed.</p>	<p>N= 1,2,...,50</p> <p>Tables 12.10-11</p>
NTPLOT(N)	<p>array of time step numbers at which information is written to the binary plot file (filename=PLTDAT, unit=76); entries may be:</p> <p><0: write at every NTPLOT th time step; no further values in array are processed;</p> <p>=0: no further writes;</p> <p>>0: write at indicated step; after this step, use the next entry in the array.</p>	<p>N= 1,2,...,25</p>

the initial long edit in a steady-state or transient calculation and by the values specified in array NTHPR for all other long edits during a calculation. As shown in Table 12.9, the entries in these arrays cause the values of selected variables to be printed for various sections of the geometry. The entries in these arrays are signed 5-digit integers. The leading digit indicates whether values are to be displayed for cells, boundary surfaces, or thermal structures. If the leading digit is '8', information will be printed for a thermal structure; the remaining digits specify which thermal structure; the resulting display is one line (or more) for each element of the specified thermal structure prototype. If the leading digit is '9', information will be printed for a boundary surface; the next two digits specify which variable is to be displayed, using the codes shown in Table 12.10; the final two digits specify which boundary surface; the values of the specified variable will be shown for each surface element which is part of the specified boundary surface. If the leading digit is neither '8' nor '9', information will be printed for cells (or faces between cells); the first two digits indicate which variable is to be printed, using the codes shown in Table 12.11; the next digit indicates which type of plane is to be displayed; the final two digits indicate the plane index number; the values of the specified variable will be shown for each cell (or cell face) in the specified plane. For all three cases, if the entry in ISTPR or NTHPR is positive, the values are printed only for the specified cell plane, boundary surface, or thermal structure; if the entry is negative, then values are printed for all cell planes, boundary surfaces, or thermal structures between (inclusively) the one indicated by this entry and the one indicated by the next entry in the ISTPR or NTHPR array. Variables are printed in the order of the entries in array ISTPR or NTHPR; up to 50 values may be entered in each array; the final entry made by the user in these arrays should be 0 if less than 50 values are entered. For a cell plane or boundary surface, the information is displayed as a two-dimensional array (i.e., J by K, I by K, or I by J) of values for cells or boundary surface elements arranged in natural geometric order. By way of example, if the user inputs

```
ISTPR=06201,-10302,10305,90607,-05201,05210,0,
NTHPR=01308,-03201,03205,90101,-90504,90507,80001,-80005,80006,0,
```

the initial long edit will contain the volume porosity for J-plane 1, the x(r)-direction surface porosity for K-planes 2 through 5, the areas of surface elements on boundary surface 7, and the fluid temperatures for J-planes 1 through 10; all other long edits will contain the x(r) component of velocity for K-plane 8, the z component of velocity for J-planes 1 through 5, the velocity for

**Table 12.10 Identification Codes (VV) for Boundary Surface Arrays as
Used in Input Arrays ISTPR and NTHPR**

VV	Array Name	Definition
01	VELBN	u_n : velocity (in direction of surface normal vector)
02	QBN	q_b : heat flux
03	MB	adjacent internal fluid cell number (1,2,...,NM1)
04	HLB	h_b : enthalpy
05	TLB	T_b : temperature
06	AREA	A_b : area
07	RLB	ρ_b : density
08	PB	P_b : total pressure
09	IJK	adjacent internal fluid cell location indicies (of the form 'IIJJK', where II is the I index, JJ is the J index, and KK is the K index)
10	---	U_b : overall coolant/wall heat transfer coefficient (as used in the duct wall energy boundary condition)
11	TURKB	k_b : turbulent kinetic energy
12	TKEDB	ϵ_b : turbulent kinetic energy dissipation rate

boundary surface 1, the temperature for boundary surfaces 4 through 7, and temperatures for thermal structures 1, 5, and 6.

The binary plot file (file name PLTDAT on UNIT number 76) is controlled by the values input using array NTPLOT in NAMELIST DATA, as shown in Table 12.9. The input for array NTPLOT is similar to that for array NTHCON (cf. Sec. 12.4): a positive entry indicates a specific time step number and a negative entry indicates a frequency at which information is written to the plot file. If the first entry in this array is 0, no information is written to the plot file. Up to 25 values may be entered in array NTPLOT; the final entry made by the user should be 0 if less than 25 values are entered. By way of example, if the user inputs

NTPLOT=5,25,-100,2000,0,

Table 12.11 Identification Codes (VV) for Internal Arrays as
Used in Input Arrays ISTPR and NTHPR

VV	Array Name	Definition
01	UL	u: x(r)-component of velocity.
02	VL	v: y(θ)-component of velocity
03	WL	w: z-component of velocity
04	HL	h: fluid enthalpy
05	TL	T: fluid temperature
06	AL	γ_v : volume porosity
07	RL	ρ : fluid density
08	P+PSTAT0	P: total pressure
09	DL	δ_p/V_f : mass residual per fluid volume
10	ALX	γ_x : x(r)-direction surface porosity
11	ALY	γ_y : y(θ)-direction surface porosity
12	ALZ	γ_z : z-direction surface porosity
13	DRDT	d ρ /dt: time rate of change of density
14	TURK	k: turbulent kinetic energy
15	QSOUR	q''' or q'''_{in} : net heat source per fluid volume
16	PSTAT0	P ₀ : reference state pressure
17	P	P-P ₀ : pressure relative to reference state
18	DDDPOT	d(DL)/d(P). (EXPLICIT: not currently used.)
19	DDDH	d(DL)/d(HL). (EXPLICIT: not currently used.)
20	TURCON	λ_t : turbulent conductivity
21	TURVIS	μ_t : turbulent viscosity
22	TKED	ϵ : turbulent kinetic energy dissipation rate

then, after completing time step number 5, the plot file will be opened, a set of header records written (containing geometry information), followed by a set of records containing the values of certain calculational variables at time step number 5; additional sets of records containing certain calculational variables will be written at time step number 25 and then every 100th time

step (i.e. time step numbers 100, 200, 300, ...); the final two entries in the array in this example (2000,0) are never used since they follow a negative entry. The "certain" variables written to the plot file are all fluid velocity components, all fluid temperatures, and all thermal structure temperatures. (The user has control over the frequency at which variables are written to the plot file but no control over which variables are written.) The plot file is only of use as input to one of the several plotting programs (cf. Appendix D) which produce graphical displays of the computed results.

ACKNOWLEDGMENTS:

As COMMIX-1AR/P is an extension to earlier versions of COMMIX, we acknowledge the liberal use of the prior COMMIX documentation in preparing this report and thank the developers of the prior COMMIX versions for their time in discussing the implementation of the new modeling described herein.

We deeply appreciate the diligent work of Betty Kinney and Jill McGregor in typing the manuscript and preparing the figures.

This work was supported by the United States Department of Energy.

REFERENCES:

1. R. N. Blomquist, P. L. Garner, and E. M. Gelbard, "COMMIX-1AR/P: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume 1: Equations and Numerics", ANL-90/45, Argonne National Laboratory, Argonne, IL (July 1991).
2. W. T. Sha, H. M. Domanus, R. C. Schmitt, J. J. Oras, and E. I. H. Lin, "COMMIX-1: A Three-Dimensional Transient Single-Phase Component Computer Program for Thermal-Hydraulic Analysis", NUREG/CR-0785 (or ANL-77-96), Argonne National Laboratory, Argonne, IL (September 1978).
3. H. M. Domanus, R. C. Schmitt, W. T. Sha, and V. L. Shah, "COMMIX-1A: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume I: Users Manual", NUREG/CR-2896 (or ANL-82-25), Vol. I, Argonne National Laboratory, Argonne, IL (December 1983).
4. H. M. Domanus, R. C. Schmitt, W. T. Sha, and V. L. Shah, "COMMIX-1A: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume II: Assessment and Verification", NUREG/CR-2896 (or ANL-82-25), Vol. II, Argonne National Laboratory, Argonne, IL (December 1983).
5. "COMMIX-1B: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume I: Equations and Numerics", NUREG/CR-4348 (or ANL-85-42), Vol. I, Argonne National Laboratory, Argonne, IL (September 1985).
6. "COMMIX-1B: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume II: User's Manual", NUREG/CR-4348 (or ANL-85-42), Vol. II, Argonne National Laboratory, Argonne, IL (September 1985).
7. "American National Standard: Programming Language FORTRAN", ANSI X3.9-1978, American National Standards Institute, Inc., New York, NY (1978).
8. Robert Siegel and John R. Howell, Thermal Radiation Heat Transfer, 2nd edition, McGraw-Hill Book Company, New York NY, 822-831 (1981).

9. Arthur G. Hansen, Fluid Mechanics, John Wiley & Sons, Inc., New York, NY, 418-433 (1967).
10. Myer Kutz, Mechanical Engineer's Handbook, John Wiley & Sons, Inc., New York, NY, 1514-1529 (1986).
11. E. H. Novendstern, "Turbulent Flow Pressure Drop Model for Fuel Rod Assemblies Utilizing a Helical Wire-Wrap Spacer System", Nuclear Engineering and Design, **22**, 19-27 (1972).
12. F. C. Engel, R. A. Markley, and A. A. Bishop, "Laminar, Transition, and Turbulent Parallel Flow Pressure Drop Across Wire-Wrap-Spaced Rod Bundles", Nuclear Science and Engineering, **69**, 290-296 (1979).
13. Chong Chiu, Neil E. Todreas, and Warren M. Rohsenow, "Turbulent Flow Split Model and Supporting Experiments for Wire-Wrapped Core Assemblies", Nuclear Technology, **50**, 40-52 (1980).
14. P. L. Garner, R. N. Blomquist, and E. M. Gelbard, "COMMIX-1AR/P: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume 3: Programmer's Guide", ANL-92/34, Argonne National Laboratory, Argonne, IL (September 1992).
15. E. Benjamin Wylie and Victor L. Streeter, Fluid Transients, McGraw-Hill Inc., New York, NY, 102-107 (1978).
16. J. F. Koenig, Argonne National Laboratory, unpublished information (November 23, 1982).
17. F. F. Chen, Argonne National Laboratory, unpublished information (April 27, 1983).
18. G. H. Golden and J. V. Tokar, "Thermophysical Properties of Sodium", ANL-7323, Argonne National Laboratory, Argonne, IL (August 1967).
19. R. K. Lo, Argonne National Laboratory, unpublished information (September 21, 1988 and August 4, 1989).
20. E. Dunn, Argonne National Laboratory, unpublished information (January 13, 1989).
21. N. B. Vargaftik, Tables on the Thermophysical Properties of Liquids and Gases, Hemisphere Publishing Corp., Washington, DC, 521-536 (1975).
22. Kuzman Raznjevic, Handbook of Thermodynamic Tables and Charts, Hemisphere Publishing Corp., Washington, DC, 299 (1976).

23. J. J. Dongarra, C. B. Moler, J. R. Bunch, and G. W. Stewart, LINPACK User's Guide, Soc. for Ind. and Appl. Math., Philadelphia, PA (1979).
24. CA-DISSPLA: User Manual, Release 11.0, Computer Associates International, Inc., Garden City, NY (1989).

APPENDIX A: INPUT GUIDE

The main part of this report (Secs. 3 through 12) have described the card-image input file by type of model; each model typically required input in more than one block of the card-image input file. This appendix describes the card-image input file by block; within each block, the input is grouped by model. The default value is shown for each input variable.

Following the input description are several appendices. The user will find the sections on (1) Error Messages, (2) Finding Holes in the Boundary, and (3) Input Variable Index most useful when using the program. The information in the other appendices is covered in other sections of this report or in other reports on the computer program.

This appendix is distributed in machine-readable form as one of the files in the transmission of the COMMIX-1AR/P program system to a site. The remainder of the pages in this appendix have been generated from a listing of that file. As such, each page shows two page numbers: (1) the number at the center of the bottom of the page is the page number within this report and (2) the number in the upper right-hand corner is the page number within this appendix; page numbers cited within the text of this appendix (e.g. the Table of Contents near the beginning and the Variable Index near the end) refer to the latter of these two page numbers. Similarly, references cited in this appendix refer to the Reference List which is near the end of this appendix, rather than to the Reference List for this report.

COMMIX-1AR/P: Input Description: Revised 08/21/92
 Applicable to code versions 900801 and 910515.
 (These are modifications of version of APR1789.)

CCCCCCC	0000000	M	M	M	M	III	X	X	1
C	O	O	MM	MM	MM	MM	I	X	X
C	O	O	M	M	M	M	M	I	X
C	O	O	M	M	M	M	M	I	X
C	O	O	M	M	M	M	I	X	X
C	O	O	M	M	M	M	I	X	X
CCCCCCC	0000000	M	M	M	M	III	X	X	1111

III	N	N	PPPPPPP	U	U	TTTTTTT
I	NN	N	P	P	U	U
I	N	N	N	P	P	U
I	N	N	N	PPPPPPP	U	U
I	N	N	N	P	U	U
I	N	NN	P	U	U	U
III	N	N	P	UUUUUUU	U	T

GGGGGGG	U	U	III	DDDDD	EEEEEEE
G	U	U	I	D	D
G	U	U	I	D	D
G	GGG	U	U	I	D
G	G	U	U	I	D
G	G	U	U	I	D
GGGGGGG	UUUUUUU	III	DDDDD	EEEEEEE	

C O M M I X - 1 A R / P

A Three-Dimensional Transient Single-Phase
Computer Program for Thermal Hydraulic Analysis
of Single and Multicomponent Systems

developed in the

Reactor Analysis Division
Argonne National Laboratory

under sponsorship of the
United States Department of Energy

contact phone

R. N. Blomquist	708-252-8423
P. L. Garner	708-252-4872
E. M. Gelbard	708-252-4831
H. M. Park	708-252-7231

at

Building 208
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439

Updated to Version 900801 (11/30/90)
Updated to Version 910515 (09/18/91)


```
+-----+  
| Preface |  
+-----+
```

COMMIX-1AR/P is based on COMMIX-1 (Ref. 1), COMMIX-1A (Refs. 2 and 3), and COMMIX-1B (Refs. 4 and 5) and contains new models and formulations which were added to the code over a period of about 5 years. These include a pump model, radiation heat transfer, boundary conditions for inlet flow as a function of radiation surface temperature and to simulate expansion cells for constant mass calculations, multiple fluid capability, conjugate gradient solution technique for momentum/mass equations, implicit coupling of thermal structures to fluid, automatic time step control, new turbulence models, and revisions to use vector computer hardware. These model changes and the associated input changes are documented in Refs. 6 and 7. These changes have been tested using the implicit solution techniques (i.e. ISYMCH=3,4,5 and IFITEN=3) only, which are the recommended solution techniques in COMMIX-1AR/P; some models (e.g pump, radiation heat transfer, automatic time step control with IDTIME=2, and turbulence with ITURKE>1) will not work when explicit solution techniques are used.

Users of previous versions of COMMIX will need to be careful when converting old input decks to run on this version. Some input sections are completely different, and others have been slightly modified. Failure to account for the revised definitions of input variables can be disastrous. All users would be well advised to follow this document carefully in setting up their problems.

Since the development of COMMIX-1AR/P is an ongoing project, we would like to encourage user feedback. If you find coding errors or modeling weaknesses or if you have requests or suggestions for features that you feel might help other users, please communicate them to us. We will include those which we feel are appropriate in future versions of the code.

```

+-----+
| Table of Contents |
+-----+

```

COMMIX-1AR/P contains modifications relative to COMMIX-1A. Input portions with new options or which contain substantive changes are indicated with "****" below.

	Page	
COMMIX-1AR/P Input Description	1	
Preface	2	
Table of Contents	3	
General Comments	5	
Some Terminology	5	
General Input Deck Structure	6	
General Comments on NAMELIST Input	7	
PROBLEM DESCRIPTION CARDS	7	
NAMELIST GEOM	8	***
Rebalancing Option	10	
Restart Option	10	***
For IGEOM=0 or IGEOM=-1	12	
For IGEOM>0	13	
BOUNDARY SURFACE IDENTIFICATION CARDS (REG cards) . .	14	
NAMELIST DATA	16	
Debugging Parameters	18	***
Time and Time Step Related Parameters	20	***
Iteration Control Parameters	22	
Boundary Condition Types	23	***
Boundary and Cell Initialization	25	
Wall Model	26	
Fluid-Solid Heat Transfer	27	***
Material Properties (Solids)	28	***
Rebalancing Option	29	
Simplified Properties Option	31	***
Transient Functions	33	
Plot File Option	34	
Printing Option	35	
Force Structures	38	
Radiative Heat Transfer Data	40	***
Pump Model Specification	41	***
Turbulence Model	42	***
Cell Integrated Wire Wrap Force	43	
Fuel Assembly Drag Model	43	
NAMELIST INPUTQ	44	
REBALANCING REGION CARDS (REBM,REBX, etc. cards) . .	46	
FLUID CELL LOCATION CARDS (HTEX cards)	47	***
NAMELIST PUMP	48	***
PUMP LOCATION CARDS (PUMX, etc. cards)	51	***
FORCE STRUCTURE LOCATION CARDS (XFOR,etc.)	52	

NAMelist STRUCT	53
THERMAL STRUCTURE PROTOTYPE CARDS	53
NAMelist T	54
NAMelist F	55
NAMelist M	55 ***
THERMAL STRUCTURE LOCATION CARDS (OUT, IN cards)	56
Example of T.S. Input for IFRAD>0	57
BOUNDARY VALUE INITIALIZATION CARDS	58
INTERNAL CELL INITIALIZATION CARDS	59
APPENDICES	61
A. BEHIND THE SCENES	61
CONTROL PARAMETERS AT A GLANCE	61
Implicit Time Advancement	62
Explicit Time Advancement	63
STEADY-STATE DEFINITION	64
ERROR MESSAGES	65
STORAGE ALLOCATION	72
CALLING SEQUENCE	75
OVERLAY STRUCTURE	80
MACHINE DEPENDENT ROUTINES	82
FINDING HOLES IN THE BOUNDARY	83
B. INPUT VARIABLE INDEX	84
C. GRAPHICAL OUTPUT CAPABILITIES	91
D. REFERENCES	93
E. USING COMMIX AT ANL	95

```
+-----+
| General Comments |
+-----+
```

The units used in COMMIX-1AR/P are SI-based (i.e. meter, kilogram, second, joule) with two exceptions: (1) temperature is [degrees Celsius] and (2) pump speed is [revolution/minute]. The units for input variables are shown in square brackets, where "[-]" denotes the quantity is dimensionless.

Default values are indicated either by an asterisk or a value in parentheses after the variable description.

Arrays are indicated by the use of a subscript following the variable name. The ranges of the subscripts are indicated in the following table. An asterisk in the 'Current Limit' column indicates that storage is allocated at execution time according to the value in the 'Range' column.

Index	Range/Array	Limit	Page
I	IMAX	99	12
IND	IMAX*JMAX IND=I+(J-1)*IMAX	100	44-45
J	JMAX	99	12
K	KMAX	99	12,44
L	PNRAD	99	40
M	PNRAD	99	40
N	NSURF	99	12,23-26
NC	NCORR	20	39
NF	NFORCE	99*	11,39
NH	NHEATC	20	27
NI	IFHTX (<=PIFHTX)	5	31-32
NM	NMATER	20	28
NP	TVAL,FVAL for trans. func.	100	33
NP	NPUMP for pumps	5	48-50
NR	NREBRT	99	29
NTF	NEND	25	33

Limits imposed by PIFHTX, PNSTRU, PNRAD, and NTSCUP may be altered by changing the appropriate PARAMETER statements in the source code and recompiling.

```
+-----+
| Some Terminology |
+-----+
```

The computational fluid volume is partitioned into a number of computational cells, each bounded by consecutive X(or R), Y(or THETA), and Z direction grid planes. Surfaces (portions of a plane or cylinder) may be defined both on the exterior and the interior bounding the computational volume. The intersection of a surface and consecutive grid planes outlines a surface element. Surfaces which coincide with a grid plane are called regular surfaces; otherwise, they are called irregular surfaces. A regular cell is one with all faces coinciding with grid planes. Irregular cells have one irregular surface element.

```

+-----+
| General Input Deck Structure |
+-----+

```

The input deck structure depends on the geometry type:

1. Box: IGEOM=0(rectangular) or IGEOM=-1(cylindrical)
2. Hex: IGEOM>0(number of pins in fuel assembly)

```

+-----+
| Box Geometry |      The geometry is defined in terms of
+-----+      the cells formed by the intersections of
                  the X(R), Y(THETA), and Z grid planes.

```

The input deck structure, as a function of ISTATE, follows:

Block Type	ISTATE= 0	1	2	3	Notes
Problem Description	O	O	O	O	
NAMelist GEOM	R	RL	RL	RL	
Boundary Surface Identification	R	N	N	N	E
NAMelist DATA	R	RL	RL	RL	
NAMelist INPUTQ	O	N	OL	N	*
Rebalancing Regions	O	N	O	N	E
Fluid Cell Locations	O	N	N	N	E
NAMelist PUMP	O	N	OL	N	*
Pump Locations	O	N	OL	N	E*
Force Structure Locations	O	N	O	N	E
NAMelist STRUCT	O	N	O	N	
Thermal Structure Prototypes	O	N	O	N	E
Thermal Structure Locations	O	N	O	N	E
Boundary Value Initializations	R	RL	RL	RL	E
Internal Cell Initializations	R	RL	RL	RL	E

Key to codes used under ISTATE and Notes in above table:

- O: Optional (depending on other flags in input stream);
- R: Required; N: Not allowed; L: Limited input allowed;
- E: An 'END' card is part of and terminates this block;
- *: If block was present at ISTATE=0, it must be present at ISTATE=2 (even if there are no changes).

```

+-----+
| Hex Geometry |      This geometry option is useful when
+-----+      analyzing hexagonal fuel assemblies.
                  Several conventions must be noted:

```

1. Axial length is along the Z-direction and one hex flat lies on the X-axis.
2. IMAX, JMAX, DX(I), and DY(J) are automatically determined by either quarter- or full-pin partitioning.
3. Surfaces are automatically defined as follows:

Surf #	Surface Location
1	Lower left diagonal in X-Y plane
2	Upper left diagonal in X-Y plane
3	Lower right diagonal in X-Y plane
4	Upper right diagonal in X-Y plane
5	Lower flat along X-axis
6	Upper flat (parallel to X-axis)
7	Entrance plane (Z=0.0)
8	Exit plane

The input deck structure is the same as for Box Geometry, except Boundary Surface Identification Cards are not allowed.

```

+-----+
| General Comments on NAMELIST Input |
+-----+

```

- 1) The first card of each NAMELIST must begin with ' &name ', with the '&' positioned in column 2. The rest of this card may contain data values (as per Note # 3).
- 2) Other cards must begin with 1 or more blanks.
- 3) Variables are entered as VARIABLE=VALUE followed by a COMMA!
- 4) Entire arrays may be input as VAR=VAL1,VAL2,...,VALn;; when using this form for multi-dimensional arrays, the values must correspond to the order imposed by the first subscript increasing the fastest (i.e. (1,1),(2,1),...,(n,1),(1,2),(2,2),...,(n,2),...,(1,m),(2,m),...,(n,m)). Single array elements may be set using VAR(n)=VALn, .
- 5) If a variable is not entered at ISTATE=0, it will assume the listed default value. For ISTATE>0, default values are read from the Restart File, so only enter values for variables which you wish to change (except as noted in this Input Guide).
- 6) Columns 2-80 may be used for entering data values.
- 7) A sequence field (e.g. cols. 73-80) is not allowed.
- 8) The final entry of each NAMELIST must be '&END'; the '&' is not restricted to col. 2 (but col. 1 must be blank).

```

*****
* PROBLEM DESCRIPTION CARDS *
*****

```

Any number of cards with user comments can precede NAMELIST GEOM. (Comments may also generally be present between an 'END ' card and the next occurrence of ' &name ' if Cols. 1-4 are blank.)

* NAMELIST GEOM * Must ALWAYS be present, although most values
***** should not be respecified at ISTATE>0.

[illegible]

```

IGEOM      0..Rectangular (X,Y,Z) box geometry. (default)
          -1..Cylindrical (R,THETA,Z) box geometry.
              Note 1: A surface must be dedicated to R=0.0 when the
              origin is present. Set KFLOW(N)=-3 and KTEMP(N)=400
              for that surface.
              Note 2: For full  $2.0 \times \text{PI}$  radian geometries J=1 and
              J=JMAX are automatically linked thus no surfaces need
              be defined at Y=0.0 and Y= $2.0 \times \text{PI}$ . For this case, if
              ISYMCH=5 you must have JMAX > 2.
          >0..Hex geometry option. Set IGEOM to the number of pins
          in the hexagonal fuel assembly. The following values
          are acceptable: 7,19,37,61,91,127,169,217,271.
              Notes: This option has not received extensive testing
              in COMMIX-1AR/P; use with caution. Internal limits
              may require IGEOM<=19 for IPART=0 and IGEOM<=61 for
              IPART<=1.

NL1      Total number of surface elements. (0)
          (Restriction:  $1 < \text{NL1} < 9999$ .)

NM1      Total number of computational cells. (0)
          Notes: Both NL1 and NM1 can be approximated by values
          larger than actually required; however, if this is
          done they must not be changed via NAMELIST GEOM
          when restarting (ISTATE>0). Storage is always
          allocated according to the values specified in the
          input rather than the minimum storage needed. The
          minimum values are printed when computed. If one
          desires to change NM1 and/or NL1, it must be done
          only at the start of a steady-state run (ISTATE=0).

```

Guidelines for choosing values for ISYMCH and IFITEN:
 The momentum and the energy calculational schemes must both be run in the same mode. That is, either both must be specified implicit or both must be specified explicit. (Variables ISETEN and ISETMO to control the frequency with which the energy and momentum calculations are performed.) An implicit scheme is recommended for all calculations and is required if ITURKE>1, IFPUMP>0, IRFAD>0, or IDTIME=2.

ISYMCH Momentum solution technique flag:

- 0..Explicit time advancement. Velocities are evaluated after all mass residues have been calculated using under-relaxation type iteration.
- 1..Explicit time advancement. Velocities are evaluated immediately with mass residue using successive over-relaxation (SOR) type iteration.
- 2..Explicit time advancement. Velocities are evaluated immediately with mass residue using selective successive over-relaxation (SSOR) type iteration.
- 3..Implicit time advancement using successive over-relaxation (SOR) type iteration.
- 4..Implicit time advancement using Jacobi type iteration.
- 5..Implicit time advancement using conjugate gradient (CG) solution technique. (default)

IFITEN Energy solution technique flag:

- 0..Explicit time advancement.
- 1..Explicit time advancement and implicit conduction using Jacobi type iteration.
- 2..Explicit time advancement and implicit conduction using successive over-relaxation (SOR) type iteration.
- 3..Implicit time advancement using successive over-relaxation (SOR) type iteration. (default)

IFHTX The maximum number of fluids (max. fluid #) used. (1)
 The maximum value allowed for IFHTX is set by PARAMETER PIFHTX in the source code; current value is 5.
 Selecting IFHTX>1 allows the inclusion of multiple fluids. The regions containing different fluids must be separated from each other by:

- a) cell surfaces having zero-permeability, or
- b) a boundary surface with zero velocity normal.

Selecting IFHTX>1 requires additional input: see KPROP, Boundary and Cell Initialization, Simplified Properties Option, and FLUID CELL LOCATION CARDS.


```

+-----+
| Rebalancing Option |
+-----+

```

IFREB 0..No user-specified-region rebalancing. (default)
 This is the recommended choice for ISYMCH=5.
 >0..Rebalancing is performed over user defined rebalancing
 regions and rebalancing surfaces. The value of IFREB
 is used to allocate storage for pointers and must be
 at least as large as the total number of cells in the
 rebalancing regions plus the total number of cells used
 to specify rebalancing surfaces. A value of 2 * NM1
 should be adequate space for most cases. The exact
 value needed will be printed in the Rebalancing
 Summary. The input defining the rebalancing regions
 i.e., the Rebalancing Option section of NAMELIST DATA
 and the REBALANCING REGION CARDS must be included at
 the start of a steady state run (ISTATE=0) when
 IFREB > 0. These two sections can be completely
 redefined at the beginning of a transient run
 (ISTATE=2) by setting NEWREB=1 in NAMELIST DATA.

Note: In addition to rebalancing over user-specified regions,
 plane-by-plane rebalancing is available and controlled
 by the variables IXREB, IYREB, and IZREB in the
 Rebalancing Option section of NAMELIST DATA.

```

+-----+
| Restart Option |
+-----+

```

IFRES 0..Restart files are not read or written. (default)
 1..Write restart file(UNIT=10,FILE=CMXNEW) at end.
 2..Read restart file(UNIT=9,FILE=CMXOLD) at beginning
 to continue a previous run.
 3..Combines IFRES=2 and IFRES=1.
 4..New case (ISTATE=0) with initial guess read from
 UNIT=9 and restart written to UNIT=10 at end.
 This option allows the use of a Restart File from
 a similar calculation (i.e. same geometry) to provide
 the initial guess for the following cell quantities:
 UL,VL,WI. Velocity components
 TL Temperature
 P Pressure minus reference state value
 This option essentially alleviates the user from
 inputting the above variables via the INTERNAL CELL
 INITIALIZATION CARDS.

Note: The "end" of a run, as used above, occurs by reaching
 any of (1) the max. timestep number, NTMAX; (2) the max.
 problem time, TIMAX; (3) the point where the CPU time
 remaining in the run after an outer iteration falls below
 TREST; or (4) boiling in Fluid#1, if IBOIL=1. (Item#3 uses
 a system-dependent routine TLEFT; see Appendix section on
 MACHINE DEPENDENT ROUTINES.) At the end of a run, a restart
 file is written (if requested) and execution terminates.
 (Restart notes continue on next page.)

(Restart notes continued from previous page.)

The restart file written by one run may be input to another run to continue the calculation (unless boiling occurred). When continuing a calculation, make sure ISTATE is set to the proper value and only include input for variables which you wish to change (or which are required); most all inputable values are properly restored from the restart file and should not be respecified by cards; the minimum input is preferred.

ITURKE Turbulence Model Selection:

- 0..No turbulence model or user-input spatially-constant turbulent viscosity and conductivity. (default)
- 1..User-input spatially-variable turbulent viscosity and conductivity.
- 10..Zero-equation turbulence model.
- 11..One-equation turbulence model.
- 12..Two-equation turbulence model.

Notes: ITURKE>1 requires ISYMCH>2 and IFITEN=3. Additional input is described under Turbulence Models in NAMELIST DATA.

LMPRNT Printing of cell number and surface number arrays (0)

- 0..Neither array is printed. (default)
- 1..Cell number array is printed.
- 2..Both arrays are printed.

Note: Specifying LMPRNT=1 or LMPRNT=2 causes excessive geometry debugging information to be printed and causes execution to terminate. This information is of little use to the casual user. Use of LMPRNT=0 is recommended.

NFORCE Number of force structures. (0) The input defining the force structures, i.e., the Force Structure section of NAMELIST DATA and the FORCE STRUCTURE LOCATION CARDS, must be included at the start of a steady-state run (ISTATE=0) when NFORCE > 0. These two sections can be completely redefined at the beginning of a transient run (ISTATE=2) by setting NEWFOR=1 in NAMELIST DATA. The maximum value allowed for NFORCE is 99.

ISTRUC 0..No thermal structures are used. (default) Do not include NAMELIST STRUCT, THERMAL STRUCTURE PROTOTYPE CARDS, or THERMAL STRUCTURE LOCATION CARDS in the input.

- 1..Indicates that thermal structures are present in the calculation, but that no T.S. input will be read from cards. (This is not a valid choice at ISTATE=0.)
- 2..The input defining the thermal structures, i.e., NAMELIST STRUCT, THERMAL STRUCTURE PROTOTYPE CARDS, and THERMAL STRUCTURE LOCATION CARDS, are to be read in this run. This a valid choice at ISTATE=0 (initialization) or ISTATE=2 (total respecification); the existing temperature distribution is lost in ALL thermal structures if ISTRUC=2 at ISTATE=2. In either case, NEWTS must be set equal to 1 in the NAMELIST DATA. [ISTRUC is set back to 1 after it is read and its value used to conform with the existing logic in COMMIX; thus ISTRUC is printed as 1 during the input processing.]

ISTBUG 0--The storage layout table is not printed. (default)
 1--The storage layout within COMMON /SPACE/ is printed.
 (See STORAGE ALLOCATION section Appendix A.)

IBSBUG 0--The BOUNDARY SURFACE SUMMARY is not printed. (default)
 1--The BOUNDARY SURFACE SUMMARY is printed after which
 execution continues. For a description of the
 BOUNDARY SURFACE SUMMARY see the section entitled
 FINDING HOLES IN THE BOUNDARY in Appendix A.
 2--The BOUNDARY SURFACE SUMMARY is printed after which
 execution terminates.

```
+-----+
| For IGEOM=0 or IGEOM=-1 the following variables |
| must be included in NAMELIST GEOM.             |
+-----+
```

IMAX The maximum number of cells in the X(R)-direction. (1)
 Restriction: $1 \leq \text{IMAX} \leq 99$.

JMAX The maximum number of cells in the Y(THETA)-direction. (1)
 Restrictions: (a) $1 \leq \text{JMAX} \leq 99$. (b) A full cylinder
 (i.e. $\text{IGEOM} = -1$ with $\text{sum DY} = 2\pi$) requires $\text{JMAX} \geq 3$
 if $\text{ISYMCH} = 5$.

KMAX The maximum number of cells in the Z-direction. (1)
 Restriction: $1 \leq \text{KMAX} \leq 99$.

NSURF The number of unique surfaces enclosing the calculational
 area. Unique surfaces are determined by a unique
 combination of the following four characteristics:
 1. Velocity Boundary Condition,
 2. Temperature Boundary Condition,
 3. The unit normal vector to the surface,
 4. Adjacent Fluid Type.
 Restriction: $1 \leq \text{NSURF} \leq 99$.

DX(I) Calculational cell sizes along the X-axis($\text{IGEOM} = 0$) [m]
 or R-axis($\text{IGEOM} = -1$) [m]; input IMAX values.

DY(J) Calculational cell sizes along the Y-axis($\text{IGEOM} = 0$) [m]
 or THETA-axis($\text{IGEOM} = -1$) [radian]; input JMAX values.

DZ(K) Calculational cell sizes along the Z-axis [m];
 input KMAX values.

The unit normal vectors referred to by the following three
 variables are those pointing INTO the calculational volume;
 input values [-] for each surface N ($= 1, 2, \dots, \text{NSURF}$).

XNORML(N) X(R)-component of the unit normal vector to surface N.
 YNORML(N) Y(THETA)-component of the unit normal vector to surface N.
 ZNORML(N) Z-component of the unit normal vector to surface N.

```

+-----+
| For IGEOM>0 the following variables |
| must be included in NAMELIST GEOM. |
+-----+

```

```

IPART      0..Quarter pin partitioning is used. (default)
            1..Full pin partitioning is used.
IWIRE      0..No wire wrap option used. (default)
            1..Smeared wire wrap option used. This option is
               suggested for low Reynolds number cases. The total
               wire wrap area and total wetted perimeter over an
               axial cross section are distributed over the cross
               section such that there are two mean hydraulic
               diameters, one for cells not adjacent to a side wall
               and one for cells adjacent to side walls. The effect
               of wire wrap induced swirl flow is ignored.
            2..Cell integrated wire wrap force option used. This
               option requires IPART=0 and input for CWIREX, CWIREY,
               and CWIREZ in NAMELIST DATA.
KMAX       The maximum number of cells in the Z-direction. (1)
            Restriction: 1<=KMAX<=99.
DZ(K)      Calculational cell sizes along the Z-axis [m];
            input values for each axial level K (=1,2,...,KMAX).
CLADOD     Fuel pin diameter [m].
PITCH      Distance between fuel pin centers [m].
WALLCL     Wall clearance (distance between pin wall and duct
            wall) [m].
WODIN      Wire wrap outside diameter [m] for all wire wraps
            except those next to the duct wall for IWIRE>0.
WODOUT     Wire wrap outside diameter [m] for wire wraps next to
            the duct wall for IWIRE>0.
CWIREI     Scale factor [-] for wire wrap force model for cells
            not adjacent to duct wall for IWIRE=2.
CWIREO     Scale factor [-] for wire wrap force model for cells
            adjacent to duct walls for IWIRE=2.
ZAT0       Axial (Z) height [m] where wire wrap is positioned
            along the positive X-axis relative to the rod center
            for IWIRE=2.
WIREP      Wire wrap pitch [m] for IWIRE=2. Positive WIREP indicates
            counterclockwise rotation when looking in the negative Z-
            direction; negative WIREP indicates clockwise rotation.

```

```

+-----+
| Orphan Input | The following variables exist in
+-----+ NAMELIST GEOM. They are associated with
            obsolete or future model options. VALUES

```

MUST NOT BE INPUT BY THE USER FOR THESE VARIABLES.

```

ALXN(I,J)  ?..Old method of setting ALX() for IGEOM>0.
ALYN(I,J)  ?..Old method of setting ALY() for IGEOM>0.
ALZN(I,J)  ?..Old method of setting ALZ() for IGEOM>0.
CWIRE      ?..Old wire coefficient.

```

```

+-----+
| End of NAMELIST GEOM Input |
+-----+

```

 * BOUNDARY SURFACE IDENTIFICATION CARDS *

This set of cards must
 be present only at the
 start of steady-state

runs (ISTATE=0) when IGEOM=0 or IGEOM=-1. When present, it
 must be terminated with a card containing 'END ' in columns 1-4.

The purpose of this set of cards is to specify a set of
 boundary surfaces which completely enclose the calculational
 region and to define any other boundary surfaces inside the
 calculational region. These interior boundary surfaces must
 completely surround a surface, a cell, or a group of cells.
 To completely surround a surface one must specify two boundary
 surfaces with normals in opposite directions. A single sided
 boundary surface is not allowed in the interior of the
 calculational region. Also be sure that all surfaces specified
 bound calculational cells. Each boundary surface is defined by
 specifying one or more BOUNDARY SURFACE IDENTIFICATION CARDS,
 each of which contains the following variables in
 FORMAT (A4,F10.3,7I4):

	NAME	AREA	IB	IE	JB	JE	KB	KE	N
NAME	REG ..The surface is a regular surface. Regular surfaces lie on grid planes.								
	IREG..The surface is an irregular surface. Irregular surfaces do not lie on grid planes.								
	END ..A card with 'END ' in columns 1-4 must terminate the BOUNDARY SURFACE IDENTIFICATION CARDS.'								
AREA	<0.0..The area of each surface element is set to its actual geometrical value, either $DX(I)*DY(J)$, $DY(J)*DZ(K)$, or $DX(I)*DZ(K)$ (or equivalent values for IGEOM=-1), whichever is appropriate.								
	>0.0..The area of each surface element is assigned the input value [m ²] of AREA.								
IB,IE JB,JE KB,KE	These six variables are the beginning and ending I-, J-, and K-indices that define a rectangular solid composed of one or more cells. The rectangular solid that defines or partially defines a surface is the one adjacent to and on the side pointed to by the surface normal. (Keep in mind that the surface normals XNORML, YNORML, and ZNORML always point into the calculational region.) The intersection of each cell and the surface defines a surface element.								
N	The surface number. All surfaces with the same combination of the following five characteristics can be assigned the same surface number:								
	<ol style="list-style-type: none"> 1. Unit normal vector to the surface, 2. Velocity boundary condition, 3. Energy boundary condition, 4. Pressure boundary condition, and 5. Adjacent fluid Type. 								

- Note 1. It is possible for two surface elements to lie in the same surface and have either the same or different surface numbers as well as for two surface elements to lie in different surfaces and have the same or different surface numbers.
- Note 2. The order of the BOUNDARY SURFACE IDENTIFICATION CARDS must be as follows:
1. All IREG cards (irregular surfaces) must precede all REG cards (regular surfaces).
 2. The surface numbers, N, of all IREG cards and REG cards must be in the order of increasing value.
- Note 3. When using cylindrical geometry (IGEOM=-1), a surface must be specified at the origin when calculational cells are bounded by the origin. When an annular region is being modeled, a surface should not be defined at the origin but rather at the boundary of the first (counting from the center) calculational cell. Set KFLOW(N)=-3 and KTEMP(N)=400 for surfaces defined at the origin.
- Note 4. When using cylindrical geometry (IGEOM=-1), with $2.0 \times \pi$ radian geometries, J=1 and J=JMAX are automatically linked, thus, no surfaces need be defined at $Y=0.0$ and $Y=2.0 \times \pi$. For this case, if ISYMCH=5 you must have $JMAX > 2$.
- Note 5. The scheme to indicate surfaces in the BOUNDARY SURFACE IDENTIFICATION CARDS is the same as that used to indicate surfaces in the BOUNDARY VALUE INITIALIZATION CARDS. (This is, however, different from the scheme used to indicate surfaces in the INTERNAL CELL INITIALIZATION CARDS.) Surface elements are indicated by the cell which is adjacent to and on the side pointed to by the surface normal.

* NAMELIST DATA * Must ALWAYS be present, although input may
***** be minimal for ISTATE>0.

ISTATE Calculational mode; may be:

- 0..Start of steady-state run. (default) Geometry and boundary and initial conditions are specified in input stream; quantities not input take defaults.
- 1..Continuation of a steady-state run. Initial conditions are read from the Restart file of a previous run in which steady-state has not yet been achieved. Limited changes may be made in the input stream.
- 2..Beginning of a transient run. Initial conditions are read from the Restart file of a previous run; desirable (although not necessary) that this previous run has achieved steady-state. Some parameters may be changed in the input stream; transient functions (i.e. TVAL,FVAL,NEND) must be input at this point.
- 3..Continuation of a transient run. Initial conditions are read from the Restart file of a previous beginning-of-transient run or continuation-of-transient run. Limited changes may be made in the input stream.

The defaults for the following three variables are: (a) 1 when ISTATE=0 and (b) 0 when ISTATE=2. These variables are ignored (effectively =0) at ISTATE=1 and ISTATE=3. An input or default value of 1 requires additional input.

NEWTs 0..No new thermal structure input is read from cards.
 1..New thermal structure information is read if ISTRUC=2.

NEWREB 0..No new rebalancing information is read from cards.
 1..New rebalancing information is read if IFREB>0.

NEWFOR 0..No new force structure information is read from cards.
 1..New force structure information is read if NFORCE>0.

The two variables below give the user some control over the frequency with which the momentum and energy calculations are performed. The need for this control might arise in cases when one of the two fields (velocity or energy) varies slowly compared to the other. The intent is to be able to perform one of the calculations (momentum or energy) every time step while performing the other only occasionally resulting in a savings of CPU time. Before selecting non-default frequencies, the user should understand the full implications of skipping a calculation. Restrictions: ISETMO=M and ISETEN=N where one of the following is true:

- (a) M=1 and N is any nonzero integer;
- (b) N=1 and M is any nonzero integer;
- (c) M<0 with N>0 and M is integral multiple of N; or
- (d) N<0 with M>0 and N is integral multiple of M.

ISETEN Energy equation calculational frequency (1); may be:
 <0..Skipped every -(ISETEN)-th time step.
 >0..Performed every ISETEN-th time step.

ISETMO Momentum equation calculational frequency (1); may be:
 <0..Skipped every -(ISETMO)-th time step.
 >0..Performed every ISETMO-th time step.

IFENER 0..No energy calculation.
 1..Energy calculation is performed. (default)
 2..Energy calculation is performed with porosity adjusted
 conduction length. This option is usually used when
 analyzing hexagonal fuel assemblies (IGEOM>0).
 Note: When energy calculation is skipped (IFENER=0 or ISETEN!=1),
 enthalpy, temperature, and density remain constant in time.

IHTWAL(99) Thermal-Structure/Fluid thermal interaction (cf. Sec.
 5.4 of Ref. 7):
 0..Explicit. (default but not recommended)
 1..Semi-explicit.
 2..Implicit. This imposes restrictions; see Note#4
 under THERMAL STRUCTURE LOCATION CARDS.
 Note: Do not input IHTWAL(99)>0 unless ISTRUC>0.

IFPROP Computational basis for Fluid#1 properties:
 0..Rigorous equation-of-state subroutines. (default)
 (Packages for sodium and water are included with the
 source. Only one of these packages may be included
 when creating an executable load module.)
 1..Simplified equation-of-state approximations.
 Faster running than IFPROP=0; requires additional
 input described under Simplified Properties Option.
 Note: IFPROP only applies to Fluid #1; higher fluid numbers,
 if used, always use "simplified" properties.

IFROD 0..No fuel pin models are used. (default)
 1..Fuel pin models are used. NAMELIST INPUTQ is
 required in input.
 2..Fuel pin models are used and a special initialization
 is done. This initialization sets pressure,
 temperature, density, enthalpy, and the Z-component
 of velocity from a solution of the coupled mass,
 moment and energy equations assuming no transverse
 velocities. NAMELIST INPUTQ is required in input.
 Restrictions: IFHTX>1: IFROD<2; ISTATE>0: IFROD<2.

IBOIL 0..Do not check for boiling. (default)
 1..Check for boiling each timestep. If boiling occurs
 execution terminates; a Restart File is written if
 IFRES=1,3, or 4. (This check can only be made for
 cells containing Fluid#1 when IFPROP=0.)


```

+-----+
| Debugging Parameters |      Several debugging flags exist
+-----+                in the code which are intended
                        primarily for use in the
developmental stages. One sometimes needs the source listing
in order to determine exactly what values are printed and under
what conditions. However, quite a few flags produce self-
explanatory output. The debug flag values conform to a
simple print hierarchy as follows:

```

Flag Value	Effect
0	No print. (default)
1	Lowest level print (minimum output).
2,3,etc.	Sequentially more prints (if available).
99	Print values for each cell; this option produces large quantities of output.
> 100	Print values for a particular cell; a flag value of IIJJKK (6-digit integer) selects cell location (II,JJ,KK).

Current debug flags and the subroutine of consequence are listed below. When a short description of the flag's effect is appropriate, it is provided.

IDLBUG	= 99	(PEQN)	Difference equation coefficients (pressure eqtn.)
	> 100	(PEQN)	as above for specific cell
IENBUG	= or > 1	(QSRAD)	Each Rad.Surf: T & Q, each step
	= or > 2	(QSRAD)	Rad.Surf: arrays to find QRNET.
	= or > 2	(TSTRUC)	Rad.Surf: T**4 avg. calc.
	= 99	(ENERGI)	Dump each cell
	= 99	(ENRBAL)	Detailed energy balance (all cells)
	> 100	(ENERGI)	Dump specific cell
	> 100	(ENRBAL)	Detailed energy balance (sp. cell)
IINBUG	= or > 1	(INITZ1)	Hex can option inlet quantities
	= or > 2	(INITZ1)	Hex can option: K, WK, etc.
ITIBUG	= or > 1	(TIMSTP)	Outer iteration summary
	= or > 1	(AMAIN)	DT change noted
	= or > 2	(ENLOOP)	DHOHT,...,ITE
	= or > 2	(ENLOOP)	max. fractional changes
	= or > 2	(MOLOOP)	Iterative quantities,ITP,fractional changes, etc.
	= or > 3	(MOLOOP)	DLMAX, cell no. before/after rebalance
	= or > 3	(SOLVEN)	Final inner iter.: DHMAX,HCONV,etc.
	= or > 3	(SOLVIT)	Convergence summary
	= or > 3	(SOLVIT)	Converged DLMAX, DCONV, MDMAX, IT
	= or > 3	(TIMSTP)	Max. K.E. & TL change (IDTIME=2)
	= -3	(INITAL)	Initial pres. & dens. , all cells
	= or > 4	(SOLVEN)	Each inner iter.: DHMAX,HCONV,etc.
	= or > 4	(SOLVIT)	Inner iteration summary(unconverged)
	= or > 5	(SOLVIT)	Pressure equation coefficients for all cells.

ITKBUG	= or > 1	(TKLOOP)	max. change of TURK during outer
	= or > 3	(TKLOOP)	TURK, all cells, each outer
	= 99	(TKENER)	TURK eq. coef.,etc; all cells
	= 99	(TKSORC)	TURK eq. source; all cells
	= 99	(VELOG)	velocity gradient; all cells
	= 99	(VISFIX)	turb. visc. calc.; all cells
	> 100	(TKENER)	TURK eq. coef.,etc.; specific cell
	> 100	(TKSORC)	TURK eq. source; specific cell
	> 100	(VELOG)	velocity gradient; specific cell
ITDEBUG	= or > 1	(TDLOOP)	max. change of TKED during outer
	= 99	(TDNER)	TKED eq. coef.,etc; all cells
	> 100	(TDNER)	TKED eq. coef.,etc.; specific cell
IMOBUG	= or > 1	(CGLOOP)	Iteration summary
	= or > 1	(AMAIN)	DT change noted
	= or > 2	(CNGRIC)	Summary of DLMAX, controlling cell, etc.
	= or > 3	(CNGRIC)	Each inner iter.: PDLIR,PDUP,ICALL
	= or > 3	(CGLOOP)	Preconditioning update
	= or > 3	(CGLOOP)	A array
	= or > 3	(TIMSTP)	Max. K.E. & TL change (IDTIME=2)
IXMBUG	= 99	(XMOMI)	See source listing
	> 100	(XMOMI)	See source listing
IYMBUG	= 99	(YMOMI)	See source listing
	> 100	(YMOMI)	See source listing
IZMBUG	= 1	(BCFLOT)	Trans. velocity for each surface/time step
	= 1	(ECFLOF)	All KFLOW=-6 surfaces at each step
	= 99	(ZMOMI)	See source listing
	> 100	(ZMOMI)	See source listing

The following five flags are input in other NAMELISTs:

The following flags are input in NAMELIST GEOM (p.11-12):

IBSBUG	= 1	(FILLM)	Boundary Surface Summary
	= 2	(FILLM)	Boundary Surface Summary and STOP
ISTBUG	= 1	(AAMAIN)	Print pointers for COMMON /SPACE/
LMPRNT	= 1	(GEOM3D)	Cell #'s and STOP
	= 2	(GEOM3D)	Cell & Surf #'s and STOP

The following flag is input in NAMELIST PUMP (p.50):

IPUBUG	= 1	(PUMPQ)	Single line summary edit
	= or > 2	(PUMPQ)	3 line summary edit
	= or > 2	(PUMPQ)	Low flowrate edit
	= or > 3	(PUMPQ)	2 line summary of trans. driving fnctn.
	= or > 4	(PUMPQ)	Dynamic torque edit
	= or > 4	(PUMPQ)	Head components edit
	= or > 5	(PUMPQ)	Pump transient function edit
	= or > 5	(PUMPQ)	Pump trans. fnctn. coefficients

The following flag is input in NAMELIST STRUCT (p.53):

ITSBUG	= 1	(INPSTR)	T.S. Control Block (once)
--------	-----	----------	---------------------------

```

+-----+
| Time and Time Step Related Parameters |
+-----+

```

TSTART Initial time [s]. (0.0) The user must re-zero this variable at the beginning of a transient run, ISTATE=2.

TIMAX The maximum time [s] of this run. (3.6E+7) Normal termination occurs after this time has been reached. TIMAX refers to the simulation or problem time and not the computer CPU time needed to run the problem.

NTMAX The maximum time step number for this run. (99999) Normal termination occurs after completion of this time step.

TREST The amount of CPU time [s] to be reserved at end of job for writing a Restart File. (20.0) The amount of CPU time remaining for the job is checked at the end of each outer iteration. If the amount of time remaining is greater than TREST another iteration is performed. If not, the Restart File is written and a normal termination occurs. (When running long jobs or when the CPU/iteration is long, a value larger than the default may be useful. This implementation depends on function TLEFT; see MACHINE DEPENDENT ROUTINES in the Appendix.)

IDTIME Time step size selection (cf. Sec. 12.3 of Ref. 7):
 0..Manual: Input DT(1), DT(2), and LASTDT.
 1..Courant-Based Automatic: Step size is computed internally as the product of the largest allowable Courant-based time step size and a user-specified variable, RDTIME. (Underlying theory is based on Explicit-mode calculations.)
 2..Change-Based Automatic (default): Selection rules: If max. fractional change in cell kinetic energy or fluid temperature (a) exceeds CHGMAX, then reduce the step size by DTFACT and repeat step; (b) is less than CHGMIN for 5 consecutive steps, then increase the size of the next step by DTFACT; (c) is between CHGMIN and CHGMAX, then do not change step size. Step size will not be allowed to exceed DT(4) and calculation will terminate abnormally if step size decreases below DT(3). If this mode is selected at ISTATE=0, the calculation will initially use DT(1) & DT(2); the automatic control mode will be activated when CHGMIN is satisfied for the first time. (The code will internally set ITAUTO=1 at this point.) This avoids the use of unnecessarily small time steps during the early portion of a steady-state calculation.
 Additional input: CHGMIN, CHGMAX, DTFACT, DT(3), DT(4).

DT(1) Time step size [s] for time steps 1 through LASTDT. (0.1)
 Input for IDTIME=0 and for initial use with IDTIME=2.

DT(2) Time step size [s] for time steps after LASTDT. (0.1)
 Input for IDTIME=0 and for initial use with IDTIME=2;
 do not input DT(2) at ISTATE=1 or 3 if IDTIME=2.

LASTDT This variable in combination with DT allows the user to change the time step size during a run. The time step size for all time steps through LASTDT is taken from DT(1). After step number LASTDT, the time step size is taken from DT(2). (99999) Input for IDTIME=0 and for initial use with IDTIME=2.

RDTIME The time step size is computed internally as the product of the largest allowable time increment given the conditions and the variable RDTIME [-]. (0.8) This value is used only if IDTIME=1.

The following 5 variables are needed for IDTIME=2:

CHGMIN Fractional change [-] over step deemed small enough to allow use of longer time step size. (0.008)

CHGMAX Fractional change [-] over step deemed so large that step must be repeated with smaller step size. (0.02)
(Must have CHGMAX > CHGMIN > 0.)

DTFACT Factor [-] used to incr/decr time step size. (2.0)
(Must have DTFACT > 1.)

DT(3) Minimum allowed time step size [s]. (1.0E-05) Calculation will terminate with an error message if step size less than DT(3) is needed to satisfy CHGMAX criterion.

DT(4) Maximum allowed time step size [s]. (1.0E+05) Calculation will continue normally using DT(4) even though CHGMIN criterion indicates that a larger step could be used.
(Must have $0 < DT(3) < DT(1 \text{ or } 2) < DT(4)$.)

NTHCON Up to ten values to specify the time step numbers to call SUBROUTINE GDCONV to calculate (a) the allowed time step size under IDTIME=1 (COURDT) and (b) the convergence criteria (DCONV & VELMAX). Acceptable NTHCON values are:
0..No further calls to GDCONV.
>0..Time step number for which GDCONV is called. After the Nth positive time step number in NTHCON has been processed, the N+1th value of NTHCON is used to determine subsequent calls to GDCONV.
<0..A value of -N indicates that GDCONV is to be called every Nth time step. No subsequent values of NTHCON are considered. (-1)
See NTPRNT and NTPLOT for examples.

+-----+
 | Iteration Control Parameters | The general definitions
 +-----+ and default values of
 control parameters are
 given in this section. For a diagram showing the loop to which
 each variable relates, see the CONTROL PARAMETERS AT A GLANCE
 and STEADY-STATE DEFINITION sections in Appendix A.

IT(1)	No. of outer iterations for time steps 1 thru LASTIT. (500)
IT(2)	No. of outer iterations for time steps after LASTIT. (500) (Should select IT big enough to allow the convergence specified by EPS3 to be obtained; generally, IT needs to be increased as DT increases. Input 1 for steady state; input >1 at beginning of a transient.)
LASTIT	Time step number at which to switch from IT(1) to IT(2) maximum outer iterations per time step. (99999)
ITMAXP	Max. # of inner iterations in pressure solution. (99)
ITMAXE	Max. # of inner iterations in energy solution. (99)
OMEGAV	Under-relaxation factor [-] for the momentum equation coefficients. (1.0) (When using SOR and OMEGAV=1.0, the equation system approaches singularity for large DT.)
OMEGAE	Under-relaxation factor [-] for the energy equation coefficients. (0.8) (When OMEGAE=1.0, the equation system approaches singularity for large DT.)
OMEGA	Relaxation factor [-] for SOR pressure solution. (1.5) (Not used for ISYMCH=5 (i.e. CG) calculations.)
RELAXE	Relaxation factor [-] for energy solution. (0.95)
EPS1	Mass Residual convergence criterion [-]. (1.0E-9)
EPS2	Mass Residue convergence criterion [kg/s/m ³]. (1.0E-11) Use: (residual)/(max.mass flow/vol + EPS2) < EPS1
EPS3	Velocity and Enthalpy convergence criterion [-]. (1.0E-6) Used in: DU,DV, and DW /max. Vel < EPS3 and DH/H < EPS3 These tests on EPS3 are applied to successive outer iterations (IT) to determine whether the time step has converged. Satisfaction of these criteria are also used for printing STEADY STATE HAS BEEN REACHED. Since these are not all-inclusive checks, the user must manually check to see that total convergence (esp. T.S. temperatures) is achieved.
EPS5	Energy convergence criterion [-]. (1.0E-9) Applied to successive inner iterations (within a single outer iteration) in Subroutine SOLVEN.

```

+-----+
| Boundary Condition Types |      All BOUNDARY SURFACES (both
+-----+                      external and internal) must
                                have a velocity, temperature
                                (or heat flux), pressure, and fluid type assigned. Input
                                values for each surface N (=1,2,...,NSURF). Restriction in
                                the following: 1<=NTF<=25.

```

KFLOW(N) Type of velocity boundary condition for surface N; may be:

- 1..Constant velocity boundary with normal velocity set from VELOC(N) or explicitly specified by the BOUNDARY VALUE INITIALIZATION CARDS. The tangential component is in effect zero. (default)
- 100+NTF..Uniform transient velocity boundary with normal velocity set from the product of the NTFth transient function and VELOC(N). Only allowed if ISTATE>1.
- 200+NTF..Alternate uniform transient velocity boundary with the normal velocity set from the product of VELOC(N) and the NTFth transient function, where the function is relative velocity versus temperature of the NTFth Radiation Surface, in degrees Celsius, rather than time. Proper coupling is obtained by NTF being BOTH the number of (1) the transient function and (2) the desired radiation surface (i.e. NTF=KRAD). Only allowed if ISTATE>1.
- 1..Continuative mass flux outlet.
- 2..Continuative velocity outlet.
- 3..Free slip boundary.
- 4..Uniform velocity outlet.
- 5..Continuative mass flow outlet.
- 6..Expansion Cell (outlet/inlet) (Sec. 4.1.1, Ref.7):
 A set of cells which is totally enclosed by zero-flow boundaries should have one (additional) cell designated as an Expansion Cell to allow for density changes. One side of the Exp. Cell should be a SURFACE, having 1 SURFACE ELEMENT, KFLOW=-6, and KTEMP=400. The opposite side of this cell must border a CELL. The other 4 sides should be SURFACES (KFLOW=-3 or KFLOW=1&VELOC=0) or be separated from other cells by zero-permeability surfaces.
 During steady state (ISTATE<2), this boundary condition is identical to specifying KFLOW=-5.
 During a transient, the mass flow across the SURFACE is calculated as for KFLOW=-5; a fixup is performed at the end of the step which effectively zeros the SURFACE flows and alters the cell porosity (AL) by an amount equivalent to the apparent surface flow; this b.c. allows the fluid mass (rather than volume) contained within the system to remain constant.
 The implementation allows the calculation to continue with AL>1 but will terminate (nicely) if AL<0.02.
 A max of 10 SURFACES may be treated using KFLOW=-6. These surfaces are flagged with an "F" in the edit of Boundary Surfaces on the global balance page and their flows are excluded from the "net convected" portions of the Mass & Energy Balances which follow.

KTEMP(N) Type of energy boundary condition for surface N; may be:

- 1..Specified constant temperature boundary with temperature set from TEMP(N) or the BOUNDARY VALUE INITIALIZATION CARDS. (default) The surface heat flux is nominally computed considering the fluid conduction but not the presence of a wall. If one wishes to account for both the fluid convection and a wall conduction, the following four variables from the Wall Model section below must be specified: IHTWAL(N), HYDWAL(N), WALLDX(N), and MATWAL(N).
- 100+NTF..Uniform transient temperature boundary with temperature set from the product of the NTFth transient function and TEMP(N) for ISTATE>1. Surface heat flux is computed with options as specified above for KTEMP(N)=1.
- 200 ..Specified constant heat flux boundary with normal heat flux set from TEMP(N) or the BOUNDARY VALUE INITIALIZATION CARDS.
- 300+NTF..Uniform transient heat flux boundary with normal heat flux set from the product of the NTFth transient function and TEMP(N) for ISTATE>1.
- 400 ..Adiabatic or zero diffusive heat flux boundary.
- 500+NTF..Duct wall temperature boundary. This boundary condition type accounts for fluid convection, thermal capacity of the wall, and the heat transfer to the surrounding atmosphere or medium. The variables in the Wall Model section below must be specified. The transient function defined by NTF is a multiplier of the volumetric heat source in the wall for ISTATE>1. If a constant volumetric heat source is desired, specify a value of 500 for KTEMP(N).
- 600+N...Recirculation boundary condition. This option specifies that the temperature of the fluid emitted from one surface is the inlet temperature condition at a second surface. The receiving surface has KTEMP=600+N, with N equal to the emitting surface number. This emitting surface must have KTEMP=400. (Example: KTEMP(1)=610 indicates that surface #10 outlet temperature defines the surface #1 inlet temp.)

KPRES(N) Type of pressure boundary condition applied to the cells adjacent and interior to surface N; may be:

- 0..Boundary pressure from PRES0(i), WPRES0(i), GRAVw, where i=KPROP(N) and w=X,Y,Z. (default)
- 1..Uniform constant pressure boundary with pressure set from PRES(N). Restriction: KFLOW(N)=-1,-2,-4,-5,-6.
- 100+NTF..Uniform transient pressure boundary with pressure set from the product of the NTFth transient function and PRES(N) for ISTATE>1. Restriction: See KPRES=1.

KPROP(N) Type of fluid (indicated by Fluid #) in cell adjacent to surface N. (1) Restriction: 1<=KPROP<=IFHTX.

```

+-----+
| Boundary and Cell Initialization |
+-----+

```

The following three variables allow easy specification of uniform velocity, temperature (or heat flux), and pressure values at boundaries. Nonuniform distributions of some items can be specified using BOUNDARY VALUE INITIALIZATION CARDS.

Input values for each boundary surface N (=1,2,...,NSURF).

```

VELOC(N)    Initial velocity [m/s] at surface N in the direction
              indicated by XNORML(N), YNORML(N), and ZNORML(N). (0.0)
TEMP(N)     Depending on KTEMP(N), either initial temperature [C]
              or heat flux [W/m^2] for surface N. (0.0)
PRES(N)     Initial pressure [Pa] for surface N. (0.0)

```

The following five arrays are used to establish a reference state pressure in each fluid cell, computed using PRES0(NI), the density at TEMP0(NI), GRAVi, and the distance from point (XPRES0(NI),YPRES0(NI),ZPRES0(NI)). Input values for each Fluid # NI (1<=NI<=IFHTX) being used.

```

TEMP0(NI)   Initial temperature [C] of cells containing Fluid #NI.
              (0.0) (Must input non-zero TEMP0(NI).)
PRES0(NI)   Initial pressure [Pa] for Fluid # NI at the reference
              point (XPRES0(NI),YPRES0(NI),ZPRES0(NI)). (1.01353E+5)
XPRES0(NI)  Fluid # NI pressure ref. point: X-coordinate [m]. (0.)
YPRES0(NI)  Fluid # NI pressure ref. point: Y-coordinate [m]. (0.)
ZPRES0(NI)  Fluid # NI pressure ref. point: Z-coordinate [m]. (0.)

```

Note: Input XPRES0 and YPRES0 in Cartesian coordinates even when using cylindrcil coordinate system (IGEOM=-1).

```

GRAVX       X-component of gravity vector [m/s^2]. (0.0)
GRAVY       Y-component of gravity vector [m/s^2]. (0.0)
GRAVZ       Z-component of gravity vector [m/s^2]. (0.0)

```

Note: Input GRAVX and GRAVY in Cartesian coordinates even when using cylindrical coordinate system (IGEOM=-1). The signs of GRAVX, GRAVY, and GRAVZ are significant.


```

+-----+
| Wall Model |      The variables in this section must be
+-----+      specified for boundary surfaces having
                  the duct-wall energy boundary condition
                  (KTEMP(N)=500+NTF); they are optional for surfaces having
                  other energy boundary conditions. Enter values for each
                  boundary surface N (=1,2,...,NSURF) where appropriate.

```

WALLDX(N) Wall thickness [m] for surface N. (1.0)
 MATWAL(N) Material type for surface N. (1) The value of this variable is used as the index NM in the Material Properties (Solids) section.
 IHTWAL(N) Heat-transfer coefficient correlation number to use between the fluid and surface N. (0) The value of this variable is used as the index NH or nh in the Fluid-Solid Heat Transfer section; allowed values are in ranges 0-20 and 51-99 depending on other input.
 Note 1: KTEMP(N)=500+NTF requires IHTWAL(N)>0.
 Note 2: When IHTWAL(N)=0, the fluid-to-surface heat-transfer coefficient, if used, is evaluated simply as the total fluid thermal conductivity divided by the fluid conduction length.
 Note 3: Array location IHTWAL(99) currently serves a DUAL ROLE in also selecting the Thermal-Structure/Fluid thermal coupling; see p. 17.
 HYDWAL(N) Hydraulic diameter or characteristic length [m] associated with surface N. (0.)

The transient volumetric heat source is given by the product the following three variables and the NTFth transient function.

WALLQS(N) Average wall volumetric heat source [W/m³] for surface N. (0.0)
 QK(K) Normalized axial distribution [-]. (1.0) CAUTIONS: Array QK is also used as a heat-source multiplier for Thermal Structures, fuel pins, and coolant; if IFROD>0, array QK must be input in NAMELIST INPUTQ rather than here.
 QIJ(I,J) Normalized radial distribution [-]. (1.0) CAUTION: Array QIJ is not input directly in NAMELIST DATA; instead, if you need to override the default values, input values using array QIN in NAMELIST INPUTQ; the values in array QIN will be used internally by the code to fill array QIJ.
 TSINK(N) Temperature [C] of medium on non-fluid side of surface N. (0.0)
 HSINK(N) Heat-transfer coefficient [W/m²/C] between surface N and medium on non-fluid side. (0.0)

```

+-----+
| Fluid/Solid Heat Transfer |      Heat transfer between
+-----+                    fluid and solid (either
                               walls at Boundary Surfaces

```

or Thermal Structures) is governed by user-input correlations in one of two forms depending on the value of IHTWAL(N) or IHT. In both cases, the heat transfer rate is defined by:

$$q = h * A * (Ts - Tf)$$

where q = heat transfer rate from solid to fluid
 h = heat transfer coefficient
 A = fluid/solid interfacial area
 Ts = temperature of the solid
 Tf = temperature of the fluid

Form#1) If $1 \leq (IHTWAL \text{ or } IHT) \leq 20$ then the heat transfer coefficient is determined from

$$h = (k / D) * Nu$$

where k = laminar thermal conductivity of fluid
 D = hydraulic diameter (user input: HYDWAL or HYD)

and the Nusselt number, Nu , is given by the following form:

$$Nu = HEATC1(NH) + HEATC2(NH) * Re^{HEATC3(NH)} * Pr^{HEATC4(NH)}$$

where Re = Reynolds number ($\rho * D * vel / \mu$)
 Pr = Prandtl number ($\mu * cp / k$)
 ρ = density of the fluid
 vel = speed of the fluid
 μ = absolute laminar viscosity of the fluid
 cp = specific heat capacity of the fluid
 $HEATCi(NH)$ = user-input coefficients.

To use Form#1 the user must input:

NHEATC Number of basic heat transfer correlations. (1)
 Input the following arrays for $1 \leq NH \leq NHEATC \leq 20$.
 The first Nusselt number expression defaults to the
 Lyon-Martinelli formulation (constant heat flux along
 a tube wall for liquid metal); the remainder are zero.
 HEATC1(NH) Nusselt number additive term [-]. Since must have
 $Nu > 0$, must input $HEATC1(NH) > 0$ to accomodate zero-flow
 conditions. (7.0,19*0.0)
 HEATC2(NH) Nusselt number multiplicative term [-]. (0.025,19*0.0)
 HEATC3(NH) Reynolds number exponent [-]. (0.8,19*0.0)
 HEATC4(NH) Prandtl number exponent [-]. (0.8,19*0.0)

Form#2) If $51 \leq (IHTWAL \text{ or } IHT) \leq 99$ then the heat transfer coefficient is determined from

$$h = HEATCU(nh, m, ijk, vel, \rho, Tf, p, cp, \mu, k, D, Ts)$$

where nh = correlation number to be used ($51 \leq nh \leq 99$)
 m = fluid cell number ($1 \leq m \leq NM1$),
 ijk = cell location indices (form: $10000 * I + 100 * J + K$)
 p = local total pressure in the fluid
 $HEATCU$ = a user-supplied function

To use Form#2 the user must supply the function HEATCU having the arguments input as shown above and return the value of the heat transfer coefficient in the function name. The code has no built-in default correlations in this range.

```
+-----+
| Material Properties (Solids) |
+-----+
```

The following equations are used to define the thermal conductivity, specific heat, and density of materials other than the fluids:

$$\begin{aligned} k &= C0K(NM) + C1K(NM)*TC + C2K(NM)*TC**2 \\ cp &= C0CP(NM) + C1CP(NM)*TC + C2CP(NM)*TC**2 \\ rho &= C0RO(NM) + C1RO(NM)*TC + C2RO(NM)*TC**2 \end{aligned}$$

where k = thermal conductivity [W/m/C]
 cp = specific heat [J/kg/C]
 rho = density [kg/m^3]
 TC = temperature [C]
 NM = material index number
 Cnm = user-input coefficients (see below)

The coefficients are indexed by values of MATWAL from the Wall Model section of NAMELIST DATA and MI from NAMELIST M of the THERMAL STRUCTURE PROTOTYPE CARDS.

NMATER	Number of materials. (0) This value must be at least as large as the largest value of MATWAL and MI. Input the following arrays for 1<=NM<=NMATER<=20.
C0K(NM)	0-th order conductivity coefficient [W/m/C] for material #NM. (0.0)
C1K(NM)	1-st order conductivity coefficient [W/m/C^2] for material #NM. (0.0)
C2K(NM)	2-nd order conductivity coefficient [W/m/C^3] for material #NM. (0.0)
C0CP(NM)	0-th order Specific heat coefficient [J/kg/C] for material #NM. (0.0)
C1CP(NM)	1-st order Specific heat coefficient [J/kg/C^2] for material #NM. (0.0)
C2CP(NM)	2-nd order Specific heat coefficient [J/kg/C^3] for material #NM. (0.0)
C0RO(NM)	0-th order density coefficient [kg/m^3] for material #NM. (0.0)
C1RO(NM)	1-st order density coefficient [kg/m^3/C] for material #NM. (0.0)
C2RO(NM)	2-nd order density coefficient [kg/m^3/C^2] for material #NM. (0.0)

The thermal conductivity and specific heat capacity are currently calculated using FORTRAN FUNCTIONS which conform to the following calling sequences:

```
k = XKSOL1(NM,TC,C0K,C1K,C2K)
cp = CPSOL1(NM,TC,C0CP,C1CP,C2CP)
```

The user may supply coding to override the existing routines to specify non-quadratic forms for these two properties.

```

+-----+
| Rebalancing Option |
+-----+

```

Large scale pressure distributions such as those which exist in an initial static state or which occur during overall velocity transients are most effectively addressed with the mass rebalancing scheme. This rebalancing is effective in reducing the number of iterations required to achieve mass convergence. Rebalancing has been implemented in two different modes which can be applied separately or in combination. Neither mode of rebalancing is recommended when ISYMCH=5.

1. Plane-By-Plane Rebalancing in the X-, Y-, or Z-direction can be applied simply by specifying the appropriate values for IXREB, IYREB, and IZREB. Only one Plane-by-plane rebalancing option can be specified.

```

IXREB      0..No X-direction plane-by-plane rebalancing. (default)
            1..Plane-by-plane mass rebalancing in the X-direction
               is performed.
IYREB      0..No Y-direction plane-by-plane rebalancing. (default)
            1..Plane-by-plane mass rebalancing in the Y-direction
               is performed.
IZREB      0..No Z-direction plane-by-plane rebalancing. (default)
            1..Plane-by-plane mass rebalancing in the Z-direction
               is performed.

```

2. User-Defined-Region Rebalancing requires the user to define rebalancing regions within the fluid domain. The regions must be chosen such that region N has neighboring cells only contained in regions N-1 and N+1. Mass leaving region N and entering region N+1 does so through rebalancing surface N. Mass leaving the last region goes into the remaining cells (there must be at least one) which are not in any rebalancing region and where no rebalancing is performed. Region 1 is restricted to having neighboring cells only in region 2. One approach to choosing rebalancing regions is to exclude all cells adjacent to exits and then group the remaining cells into as many rebalancing regions as possible. Another guideline is to put rebalancing surfaces between regions of grossly different pressures. In each rebalancing region the pressure is adjusted uniformly in such a way to force the net mass nonconservation to vanish. If user-defined-region rebalancing is desired, IFREB must be assigned an appropriate positive value in NAMELIST GEOM and REBALANCING REGION CARDS must be supplied. Additionally, the following group of variables must be defined.

```

NREBRT      Number of user-defined rebalancing regions. (0)
            A maximum value of 99 is allowed for NREBRT.
NREBM(NR)   Number of cells in rebalancing region NR. (0)
            Rebalancing regions are generally chosen in such a way
            that all the cells in a given region have nearly equal
            pressure. Also, a cell may not be included in more than
            one rebalancing region.
NREBX(NR)   Number of internal X-surfaces between region NR & NR+1. (0)
NREBY(NR)   Number of internal Y-surfaces between region NR & NR+1. (0)
NREBZ(NR)   Number of internal Z-surfaces between region NR & NR+1. (0)
            (Rebalance input continues on next page.)

```

Rebalance-cont.

Interfaces between rebalancing regions often correspond to a physical structure which might cause a large pressure change.

The frequency at which rebalancing occurs is specified by the following variable.

IREBIT Rebalancing is performed before every IREBITth inner iteration if ISYMCH=3. For computational efficiency, the maximum number of iterations should be one less than a multiple of IREBIT; i.e. ITMAXP=(N*IREBIT)-1. Rebalancing is also performed before the first inner iteration if ISYMCH>=3. (50)

```

+-----+
| Simplified Properties Option |      To be included if and
+-----+                        only if IFHTX>1 in
NAMELIST GEOM or IFPROP=1 in NAMELIST DATA.

```

The following linear equations are used to approximate the state equation of Fluid #1 when the Simplified Property Option (IFPROP=1) is selected and always used for Fluid #2 thru Fluid #IFHTX:

```

Enthalpy [J/kg]      : h = FC0H(NI) + FC1H(NI) * TC
Density [kg/m^3]     : rho = FC0RO(NI) + FC1RO(NI) * TC
or rho = (P * GASMW(NI)) / (8314. * (TC + 273.16))
Conductivity [W/m/C] : k = FC0K(NI) + FC1K(NI) * TC
Viscosity [Pa*s]     : mu = FC0MU(NI) + FC1MU(NI) * TC

```

where TC is the temperature [C], P is the pressure [Pa], and NI is the Fluid #. The form used for density is selected based on the values input for FC0RO and GASMW; only one of these two may be non-zero for each fluid. Input values in the following coefficient arrays for Fluid #1 if IFPROP=1 and for all other fluids actually being used in the range $2 \leq NI \leq IFHTX$.

```

FC0H(NI)  0-th order enthalpy coefficient [J/kg] for Fluid #NI. (0.0)
FC1H(NI)  1-st order enthalpy coefficient [J/kg/C] for Fluid #NI. (0.0)
FC0RO(NI) 0-th order density coefficient [kg/m^3] for Fluid #NI. (0.0)
           (FC0RO non-zero selects linear density fit.)
FC1RO(NI) 1-st order density coefficient [kg/m^3/C] for Fluid #NI. (0.0)
GASMW(NI) Gas molecular weight [kg/kgmole?] for Fluid #NI. (0.0)
           (GASMW non-zero selects ideal gas law type density.)
FC0K(NI)  0-th order conductivity coefficient [W/m/C] for Fluid #NI. (0.0)
FC1K(NI)  1-st order conductivity coefficient [W/m/C^2] for Fluid #NI. (0.0)
FC0MU(NI) 0-th order viscosity coefficient [Pa*s] for Fluid #NI. (0.0)
FC1MU(NI) 1-st order viscosity coefficient [Pa*s/C] for Fluid #NI. (0.0)

```

To allow the user to check properties, the code will print a small table for each Fluid #NI actually being used with temperature values ranging from FCTLO to FCTHI, in 10 equal-increment steps, showing properties at PRES0(NI). The default values shown below depend on which detailed fluid property package has been installed and are shown in the order (sodium; light water; heavy water; helium).

```

FCTLO      Lower temperature limit [C] for table. (300.0; 20.0;
           0.0; 0.0)
FCTHI      Upper temperature limit [C] for table. (700.0; 100.0;
           200.0; 1000.0)

```

----- THIS PAGE IS CURRENTLY BLANK -----

```
+-----+
| Transient Functions |
+-----+
```

All transient driving functions are input using arrays TVAL, FVAL, and NEND. They MAY be input at any point in the calculation; they MUST be input at the beginning of a transient (ISTATE=2) even if they have been input previously. To respecify at ISTATE>0, input data for ALL functions, not just the ones to be changed. Each function is defined by a user-specified set of points, using TVAL and FVAL. The entries in TVAL for each function must be non-decreasing. In both the TVAL and FVAL arrays, the first value of the second function immediately follows the last value of the first function. The same pattern must be followed for all subsequent functions. Make sure that the range input for TVAL completely covers the range to be used, since the code does not extrapolate. Cubic spline fit coefficients are then generated by the code for each range between successive TVAL entries for each function; entering the same TVAL twice in succession causes the derivative of the resultant fits to be discontinuous; if the corresponding FVAL values are different, the resultant fits will also be discontinuous in value. Ten to fifteen points per function, with points concentrated at rapidly changing FVAL values, are normally sufficient; however, the user is cautioned to carefully check the fit summary printed by the code for unwanted abnormal behavior. The TOTAL number of points allowed for the specification of ALL transient functions (i.e. NP below) is 100. The maximum number of functions (i.e. NTF below) is 25.

TVAL(NP) The independent variable for the transient functions. The independent variable is (a) temperature [C] for KFLOW=200+NTF and (b) time [s] for all others.

FVAL(NP) The dependent variable [-] at TVAL(NP) for the transient functions.

NEND(NTF) The number of points in the NTFth transient function. The maximum value allowed for NTF is 25. The sum of the entries in the NEND array must be <= 100. (If changing the number of transient functions at ISTATE>0, then must input NEND(max(NTF)+1) as 0.)

NTOTS(NTSP) Transient Function # for heat source in Thermal Structures. The values in NTOTS are input in the same order as Thermal Structure Prototypes are defined (and, thus, index NTSP does not necessarily correspond to the T.S. Prototype index number N) and override any values of NT specified (during this run or previously) on T.S. Prototype Cards. (NTOTS input is ignored at ISTATE=1 and =3.) Include this input only if the the values of NT are to be overridden.

NOFQT Number of the transient function which is used as a multiplier of the heat source for the coolant when thermal structures are present and as a multiplier of total heat source when no thermal structures are present. (0) (See NAMELIST INPUTQ for more information.)


```

+-----+
| Plot File Option |
+-----+

```

NTPLOT(i) Time step numbers at which plotting information is written to the Plot file (UNIT=76,FILE=PLTDAT). (25*0)
 Each entry i (=1,2,...,25) may be:
 0..No more information is written. (default) If
 NTPLOT(1)=0, the file is never opened.
 >0..Time step number at which information is written.
 After this write, the next value in NTPLOT is used
 to determine the next time step number at which to
 write information.
 <0..Information is written at time step numbers which
 are an integral multiple of |NTPLOT|. Subsequent
 entries in NTPLOT are not considered.

Examples:

```

NTPLOT=-5 ..every 5th step is written.
NTPLOT=5,30,-20 ..steps 5, 30, 40, 60, etc. are written.
NTPLOT=10,20,0 ..only steps 10 and 20 are written.

```

This file contains an initial block of geometry information. Then, for every time step specified by NTPLOT, a block is written which contains all fluid velocities (UL, VL, WL, and VELBN), all fluid and structure temperatures (TL, TLB, and TTS), and the volumetric heat source (QSOUR) for all cells. (This same information for the last time step in a job is also appended to the Restart File (UNIT=10).) In a subsequent job step, the user may invoke one of several programs to display selected results graphically in a variety of formats. See Appendix C: GRAPHICAL OUTPUT CAPABILITIES for further information.

```

+-----+
| Printing Option |      Calls to SUBROUTINE OUTPUT are
+-----+      controlled by the two variables
                        NTPRNT and TPRNT. They can be used
individually or together. The information printed at each
call to SUBROUTINE OUTPUT is determined by the variables
ISTPR and NTHPR described below.

```

NTPRNT(i) Time step numbers at which SUBROUTINE OUTPUT is to be called. (-9999,48*0) Each entry i (=1,2,...,49) may be:

- 0..No more calls to SUBROUTINE OUTPUT. In a continuation run, previously specified NTPRNT values may be overridden by specifying the desired new values followed by ",0," in NTPRNT.
- >0..Time step number at which SUBROUTINE OUTPUT is to be called. After this call, the next value in NTPRNT is used to determine the next time step number at which to call OUTPUT.
- <0..SUBROUTINE OUTPUT is called at time step numbers which are an integral multiple of |NTPRNT|. Subsequent entries in NTPRNT are not considered.
- 9999..SUBROUTINE OUTPUT is called just before the run is terminated. (default)

Examples:

```

NTPRNT=0 ..after initialization, OUTPUT is never called.
NTPRNT=5,10,-9999 ..call OUTPUT at steps 5 and 10, and
just before termination.

```

TPRNT(i) Problem times [s] at which SUBROUTINE OUTPUT is to be called. (50*0.0) OUTPUT is called at the time step at which the problem time equals or first exceeds each print time as specified in TPRNT. Each entry i (=1,2,...,50) may be:

- 0.0..No more calls to SUBROUTINE OUTPUT. (default) In a continuation run, previously specified values of TPRNT may be overridden by specifying the desired new values followed by ",0.0," in TPRNT.
- >0.0..Time at which SUBROUTINE OUTPUT is to be called. After this call, the next value in TPRNT determines the next time at which to call OUTPUT.
- <0.0..SUBROUTINE OUTPUT is called at times which are an integral multiple of |TPRNT|; the next entry in TPRNT is the time (>=0.0) at which to begin calling OUTPUT at this frequency. Subsequent entries in TPRNT are not considered.

Examples:

```

TPRNT=1.0,5.0,-10.0 ..call OUTPUT at (or after) times 1.0,
5.0, 10.0, 20.0, . . etc..
TPRNT=-5.0,10.0 ..call OUTPUT at (or after) times 10.0,
15.0,20.0,. . . etc..

```

ISTPR(i) Specification of which portions of which arrays are to be printed in the first call to SUBROUTINE OUTPUT in a run having ISTATE=0 or 2. (50*0)

NTHPR(i) Specification of which portions of which arrays are to be printed by SUBROUTINE OUTPUT for all calls except the first call in a run having ISTATE=0 or 2. (50*0)

The format of each entry i (=1,2,...,50) for ISTPR and NTHPR depends on whether internal arrays, surface arrays, or thermal structures are to be displayed.

For internal arrays, each value of ISTPR and NTHPR is a signed five-digit integer of the form 'SVVPLL' which is coded according to the following rules:

S +...Only the plane specified by 'VVPLL' is printed.
 (A plus sign is assumed and need not be specified.)
 -...All planes between the values of 'LL' on the current
 and following values of ISTPR or NTHPR are printed.

VV 01..UL X(R)-component of velocity.
 02..VL Y(THETA)-component of velocity.
 03..WL Z-component of velocity.
 04..HL Fluid enthalpy.
 05..TL Fluid temperature.
 06..AL Volume porosity.
 07..RL Fluid density.
 08.. Total dynamic pressure (P+PSTAT0).
 09..DL Residual mass.
 10..ALX X(R)-direction surface permeability.
 11..ALY Y(THETA)-direction surface permeability.
 12..ALZ Z-direction surface permeability.
 13..DRDT d(RL)/d(TIME).
 14..TURK Fluid turbulent kinetic energy.
 15..QSOUR Fluid volumetric heat source.
 16..PSTAT0 Reference state (hydrostatic) pressure.
 17..P Pressure relative to PSTAT0.
 { 18..DDDPOT d(DL)/d(P). (EXPLICIT-MODE ONLY) }
 { 19..DDDH d(DL)/d(HL). (EXPLICIT-MODE ONLY) }
 20..TURCON Fluid turbulent conductivity.
 21..TURVIS Fluid turbulent viscosity.
 22..TKED Fluid turb. kinetic energy dissipation rate.

P 1..An I-plane is printed.
 2..A J-plane is printed.
 3..A K-plane is printed.

LL Specific plane to be printed. If S is +, only one plane is indicated. If S is -, the 'LL' values in the current and next values of ISTPR or NTHPR indicate the range of planes to be printed.

(ISTPR/NTHPR input continues on next page.)

(ISTPR/NTHPR input continued from previous page.)

For surface arrays, each value of ISTPR and NTHPR is a signed five digit integer of the form 'S9VVLL' which is coded according to the following rules:

- S +...Only surface number 'LL' is printed.
 (A plus sign is assumed and need not be specified.)
 -...All surfaces between the values of 'LL' in the
 current and following values of ISTPR or NTHPR are
 printed.
- VV 01..VELBN Normal surface velocity.
 02..QBN Surface heat flux.
 03..MB Adjacent internal cell number.
 04..HLB Surface enthalpy.
 05..TLB Surface temperature.
 06..AREA Surface element area.
 07..RLB Surface density.
 08..PB Surface pressure.
 09..IJK Adjacent internal cell indices. Each value is
 of the form 'IIJJKK' where II is the I index,
 JJ is the J index, and KK is the K index.
 10.. Overall heat transfer coefficient from coolant
 to surface (QBN(L)/(TLB(L)-TL(MB(L)))).
 11..TURKB Surface turbulent kinetic energy.
 12..TKEDB Surface turb. kinetic energy dissipation rate.
- LL Specific surface to be printed. If S is +, only one
 surface is indicated. If S is -, the 'LL' values in the
 current and next values of ISTPR or NTHPR indicate the
 range of surfaces to be printed.

For thermal structure information, each value of ISTPR and NTHPR is a signed five digit integer of the form 'S8NNNN' which is coded according to the following rules:

- S +...Only structure number 'NNNN' is printed.
 (A plus sign is assumed and need not be specified.)
 -...All structures between the values of 'NNNN' in the
 current and following values of ISTPR and NTHPR are
 printed.
- NNNN Specific structure to be printed. If S is '+', only one
 structure is indicated. If S is '-', the 'NNNN' values
 in the current and next values of ISTPR or NTHPR indicate
 the range of surfaces to be printed. ("NNNN" refers
 to the "N=" entry in the T.S. Prototype definitions.)

Example:

ISTPR=06105,-10301,10305,

NTHPR=01308,-03201,03205,-90504,90507,80001,-80005,80006,

The first call to OUTPUT will print the volume porosity for I-plane 5 and the x(r)-direction surface porosity for K-planes 1 through 5. All subsequent calls to OUTPUT will print the x(r)-component of velocity for K-plane 8, the z-component of velocity for J-planes 1 through 5, the temperature of boundary surfaces 4 through 7, and information for thermal structures 1, 5, and 6.

```

+-----+
| Force Structures |
+-----+

```

The Force Structure parameters are required only when NFORCE>0 in NAMELIST GEOM. (Force Structure locations are specified using FORCE STRUCTURE LOCATION CARDS.)

The Force Structure is a mechanism whereby a force can be applied to the fluid across a cell face between two computational cells. This force, expressed as a pressure gradient [Pa/m], is computed using one of the following:

```

DPDX=-FORCEF(NF)*RL*ABS(UL)*UL*FCORR(Re)/CLENTH(NF)
DPDY=-FORCEF(NF)*RL*ABS(VL)*VL*FCORR(Re)/CLENTH(NF)
DPDZ=-FORCEF(NF)*RL*ABS(WL)*WL*FCORR(Re)/CLENTH(NF)

```

where FCORR = friction correlation (function of Re)
 Re = $RL \cdot \sqrt{UL^2 + VL^2 + WL^2} \cdot REYLEN(NC) / VIS$
 RL = local fluid density
 UL, VL, and WL = local fluid velocity components
 VIS = local fluid viscosity
 FORCEF, CLENTH, and REYLEN = user input (see below)

Normally, FCORR is given by the general forms:

```

FCORR=ACORRL(NC)*Re**BCORRL(NC)+CCORRL(NC)
      when      Re < REYTRN(NC), and
FCORR=ACORRT(NC)*Re**BCORRT(NC)+CCORRT(NC)
      when      Re >= REYTRN(NC)

```

where the user inputs the constants (A,B,C)CORR(L,T) and REYTRN; the user may define up to twenty such forms, which correspond to ICORR values of 1 to 20.

Alternately, FCORR may be specified completely by the user in subroutine FORCES for ICORR values of 50 through 99. Correlations are already present in this subroutine for ICORR=81-87 and 90-95, for which REYLEN and CLENTH are also defined and may not be overridden via card input here. (App. C of Ref. 5 has a collection of resistance correlations.)

(Force structure input continues on next page.)

(Force structure input continued from previous page.)

The user must input the following for each force structure number NF (=1,2,...,NFORCE):

ICORR(NF) The correlation type of force structure NF.
 1-20..refers to normal correlations.
 User supplies (A,B,C)CORR(L,T),REYTRN.
 21-49..not allowed.
 50-99..refers to alternates coded in subroutine FORCES.
 The following may be used without changing FORCES:
 81..IFR Fuel Assembly
 82..IFR Blanket Assembly
 83..IFR Removable Shield/Reflector Assembly
 84..Flow in a Pipe
 85..IHX (currently same as #84)
 86..DRACS (currently same as #84)
 87..Stopped centrifugal pump rotor
 90..CRBR fuel assembly
 91..CRBR blanket assembly
 92..DRHX (Direct Reactor Heat Exchanger)
 93..CRBR chimney assembly
 94..FFTF pin bundles
 95..CRBR control assembly

FORCEF(NF) Force coefficient [-] for force structure NF.

REYLEN(NF) Length [m] used to compute the Reynolds number for force structure NF.

CLENTN(NF) Characteristic length [m] used in above DPDi equation.
 >0.0..The value input is used.
 <0.0..A characteristic length computed from either
 DX, DY, or DZ, whichever is appropriate, is used.

Note: REYLEN and CLENTN are not used for ICORR=81-87,90-95.

If any value of ICORR <=20 then also input the following:

NCORR The number of correlation types to be input via the following variables (1<=NC<=NCORR<=20).

REYTRN(NC) Reynolds number [-] for laminar/turbulent transition.
 (Numerical instability may occur if this value is far from the actual crossover point between the ranges.)

ACORRL(NC) Correlation coefficients [-] in FCORR equation when
 BCORRL(NC) the Reynolds number (Re) is less than REYTRN(NC) (i.e.
 CCORRL(NC) laminar flow).

ACORRT(NC) Correlation coefficients [-] in FCORR equation when
 BCORRT(NC) the Reynolds number (Re) is greater than REYTRN(NC)
 CCORRT(NC) (i.e. turbulent flow).

```

+-----+
| Radiative Heat Transfer Data |
+-----+

```

The Radiative Heat Transfer (PHT) data are required when IFRAD > 0 (see below). The RHT methodology is described in Sec. 6.4 of Ref. 6. The model is activated by specifying the items below plus KRAD in NAMELIST M of the thermal structure surface which is radiating. The implicit energy calculational scheme (IFITEN=3) must be used if IFRAD>0.

IFRAD 0..No Radiative Heat Transfer (default)
 1..For a radiative calculation in which the number
 of solid surfaces (i.e. thermal structures) equals
 the number of radiating surfaces. (Not functional;
 always use IFRAD=2 to include radiation.)
 2..For a radiative calculation in which the number
 of solid surfaces does not equal the number of
 radiating surfaces (e.g. when a thermal structure
 radiates from both sides).

NRAD L..The total number of radiating surfaces in the
 calculational domain. If IFRAD=1, the code will
 compute the value internally; when IFRAD=2, the
 user must input this value. (The maximum allowed
 value for L is set by PARAMETER PNRAD in the
 source code and currently has a value of 99.)

EMITS(L) Emissivity [-] of radiating surface # L.

FVIEW(L,M) View factors [-]: defined as the fraction of energy
 leaving surface L that arrives on surface M.
 The value of FVIEW is always between 0.0 and 1.0.
 The view factor array is sized as PNRAD x PNRAD
 (see NRAD), of which NRAD x NRAD values pertain
 to the calculation at hand. Those array elements
 beyond the relevant range of NRAD x NRAD must be
 input as zeros. The user should insure that
 the values input obey: (1) reciprocity:
 $FVIEW(L,M) \cdot area(L) = FVIEW(M,L) \cdot area(M)$ and
 (2) sum of FVIEW(L,m) over all m is 1 for each L.

OMR The under-relaxation factor [-] used in the
 calculation of the radiating heat flux:
 $q = OMR * q(new) + (1-OMR) * q(old)$
 The iteration is quite sensitive at the
 beginning of a calculation: small values for
 OMR are recommended at that point. (0.20)

```

+-----+
| Pump Model Specification |
+-----+

```

Use of pumps is indicated by the IFPUMP flag. Selecting IFPUMP>0 requires additional input: NAMELIST PUMP and PUMP LOCATION CARDS; also, an "implicit" calculational scheme (ISYMCH>=3) must be used. Although there may be multiple pumps (up to 5), all pumps must use the same model type.

```

IFPUMP      Pump model type selection flag; may be:
            0..No pump modeling (default);
            1..Specified Speed Model;
            2..Specified Pressure-Increase Model; allowed in
               transient calculations (ISTATE>=2) only;
            3..Homologous Model: pump operational over all
               four quadrants;
            4..Homologous Model: reverse pump operation is
               precluded to represent rotor lock.

```



```

+-----+
| Turbulence Modeling |
+-----+

```

In all of the following turbulence models, an effective viscosity is used in the diffusion term of the momentum equation. This effective viscosity is the sum of the turbulent viscosity and the molecular viscosity. Similarly, an effective thermal conductivity is used in the diffusion term of the energy equation which is likewise the sum of the turbulent thermal conductivity and the molecular thermal conductivity. See Sec. 7 of Ref. 6 for more details.

The type of input required depends on the value input for ITURKE in NAMELIST GEOM; there are five categories. CAUTION: The code uses the following input constants for all fluids present; the validity of this is being checked.

```

+-----+
| 1) ITURKE=0:                               |
| Constant Turbulent Diffusivity Model |
+-----+

```

The turbulent viscosity and turbulent conductivity are constant everywhere. Input the following:

TURBV	Turbulent viscosity [Pa*s]. (0.0) If TURBV is input as (or defaulted to) 0, then there is no turbulence modeling. The constant turbulence model is activated by inputting (i) TURBV>0 and either (ii.a) TURBC or (ii.b) CHARRE and CHART.
TURBC	Turbulent conductivity [W//m/C]. (0.0) Turbulent conductivity can be specified directly in TURBC or by omitting TURBC and specifying values for CHARRE and CHART to be used in the following: $\text{TURBC} = \text{Cp} * \text{TURBV} / (.8 / (1 - \exp(-6.E-5 * \text{CHARRE} * \text{Pr}^{.333})))$ where Cp and Pr are the fluid specific heat capacity and Prandlt number for Fluid #1 evaluated at CHART and PRES0(1).
CHARRE	Characteristic Reynolds number [-]. (0.0)
CHART	Characteristic temperature [C]. (0.0)

```

+-----+
| 2) ITURKE=1: User Override of             |
| Constant Turbulent Diffusivity Model |
+-----+

```

Same as case 1 but the turbulent viscosity and conductivity as initialized by TURBV and TURBC may be overridden on a cell by cell basis. Input the items shown for ITURKE=0 and provide overrides using 'TURV' and TURC' records in INTERNAL CELL INITIALIZATION CARDS input block.

```

+-----+
| 3) ITURKE=10:                |
| Zero-Equation Turbulence Model |
+-----+

```

This option does not solve any governing equations involving turbulent quantities. The turbulent viscosity and thermal conductivity are computed from the following equations:

$$\text{TURVIS} = \rho \cdot \text{lensca}^{**2} \cdot \text{velog}$$

$$\text{TURCON} = \text{TURVIS} \cdot \text{cp} / \text{PRNDLH}$$

where ρ = local fluid density,
 cp = local fluid specific heat capacity,
 lensca = length scale ($0.42 \cdot \text{ydist}$),
 ydist = distance to the nearest wall (the code uses
 a cutoff value of $0.175 \cdot \text{HYDIN}$),
 velog = velocity gradient, and
 PRNDLH = user-input coefficient.

Input the following:

HYDIN D_h : Hydraulic diameter [m]. If $\text{IGEOM} > 0$, HYDIN is computed internally; otherwise, user must input non-zero value.
 PRNDLH σ_h : Turbulence Prandtl number [-] used to compute turbulent conductivity, TURCON (see eq. above), for thermal energy transfer. (0.9)
 OMEGAT OMEGA_t : Relaxation factor [-] for turbulent viscosity. (0.7)

```

+-----+
| 4) ITURKE=11:                |
| One-Equation Turbulence Model |
+-----+

```

The equation for turbulent kinetic energy (TURK) is solved. Wall function corrections are applied to cells adjacent to solid walls for both the turbulent kinetic energy equation and the momentum equations. The turbulent viscosity and thermal conductivity are computed from the following equations:

$$\text{TURVIS} = \text{CDTURB} \cdot \rho \cdot \text{TURK}^{**2} / \text{TKED}$$

$$\text{TURCON} = \text{TURVIS} \cdot \text{cp} / \text{PRNDLH}$$

where ρ = local fluid density,
 cp = local fluid specific heat capacity,
 TURK = local turbulent kinetic energy,
 TKED = dissipation of turbulent kinetic energy
 computed from the following equation:
 $\text{TKED} = \text{CDTURB}^{**0.75} \cdot \text{TURK}^{**1.5} / (\text{AKAPPA} \cdot \text{ydist})$
 AKAPPA = $0.23 / (\text{CDTURB}^{**0.25})$,
 ydist = distance to the nearest wall (the code uses
 a cutoff value of $0.175 \cdot \text{HYDIN}$),
 PRNDLH = user-input coefficient, and
 CDTURB = user-input coefficient.

Input the items under ITURKE=10 plus the following:

CDTURB C_{\mu}: Coefficient [-] for computation of turbulent viscosity. (0.09)

TDIN C_{\epsilon, in}: Coefficient [1/m], in the following equation which is used to compute the initial and inlet turbulent kinetic energy dissipation rate: $TKED_{in} = TDIN * TURK^{**1.5}$. If empirical data are not available, an estimate of TDIN may be obtained from: $TDIN = CDTURB / (0.04025 * HYDIN)$. (User must input a non-zero value for TDIN.)

TKIN C_{k, in}: Coefficient [-] in the following equation which is used to compute initial and inlet turbulent kinetic energy: $TURK = TKIN * velocity^{**2}$. (0.001)

PRNDLK sigma_k: Turbulence Prandtl number [-] used to compute an effective viscosity, $visc = visc(molecular) + TURVIS / PRNDLK$, for use in turbulent kinetic energy transport. (1.0)

NTJUMP Frequency with which the turbulence subroutines are called; i.e. every NTJUMP-th time step. (1)

ITMAXK Maximum number of inner iterations for solution of the transport equations for turbulent kinetic energy and, if ITURKE=12, its dissipation rate. (99)

EPS6 epsilon_k and epsilon_{\epsilon}: Inner iteration convergence criterion [-] for solution of the transport equations for turbulent kinetic energy and, if ITURKE=12, its dissipation rate. (1.0E-9)

OMEGAK OMEGA_k: Relaxation [-] factor for the turbulent kinetic energy equation coefficients. (0.7)

RELAXK omega_k and omega_{\epsilon}: Relaxation factor [-] for solution of the transport equations for turbulent kinetic energy and, if ITURKE=12, its dissipation rate. (0.8)

```

+-----+
| 5) ITURKE=12: |
| Two-Equation Turbulence Model |
+-----+

```

This is the most rigorous turbulence model in the code. Both the equation for turbulent kinetic energy (TURK) and the equation for dissipation of turbulent kinetic energy (TKED) are solved. Wall function corrections are applied to cells adjacent to solid walls for both the turbulent kinetic energy equation and momentum equations. The turbulent viscosity and thermal conductivity are computed using the following equations:

$$TURVIS = CDTURB * \rho * TURK^{**2} / TKED$$

$$TURCON = TURVIS * cp / PRNDLH$$

where ρ = local fluid density,
 cp = local fluid specific heat capacity,
 $TURK$ = local turbulent kinetic energy,
 $TKED$ = dissipation of turbulent kinetic energy,
 $PRNDLH$ = user-input coefficient, and
 $CDTURB$ = user-input coefficient.

Input the items under ITURKE=11 (and 10), except that HYDIN is not used, plus the following:

CT1 C_1: Empirical constant [-] in source term (Eq. 7.9 of Ref. 6) of transport equation for turbulent kinetic energy dissipation rate. (1.44)

CT2 C_2: Empirical constant [-] in source term (Eq. 7.10 of Ref. 6) of transport equation for turbulent kinetic energy dissipation rate. (1.92)

OMEGAD OMEGA_d: Relaxation factor [-] for the turbulent kinetic energy dissipation rate equation coefficients. (0.7)

PRNDLD sigma_(epsilon): Turbulence Prandtl number [-] used to compute an effective viscosity, $\text{visc} = \text{visc}(\text{molecular}) + \text{TURVIS}/\text{PRNDLD}$, for use in turbulent kinetic energy dissipation rate transport equation. (1.3)

```
+-----+
| Cell Integrated Wire Wrap Force |      These parameters are
+-----+                          used only when IWIRE=2.
```

CWIREX Coefficient [-] of wire force in X-direction. (0.5)

CWIREY Coefficient [-] of wire force in Y-direction. (0.5)

CWIREZ Coefficient [-] of wire force in Z-direction. (0.5)

```
+-----+
| Fuel Assembly Drag Model |
+-----+
```

This model provides additional flow resistance and is intended for use with the hexagonal fuel assembly option (IGEOM>0).

IDRAG 0..No flow resistance due to fuel assembly drag model (*).

1..Nominal fuel assembly drag forces applied.

2..Fuel assembly drag forces are applied as in option 1 except that the cross flow drag is multiplied by AL/ALX or AL/ALY where AL is the volume porosity and ALX and ALY are surface permeabilities. This option should be used when fuel assembly drag forces are desired and the cell integrated wire wrap force option is used (IWIRE=2).

Restriction: IDRAG>0: IFROD>1, IMAX<=10, JMAX<=10 (due to dimensions of array WETLEN).

CDRAGX Multiplier [-] of drag force in X-direction. (1.0)

CDRAGY Multiplier [-] of drag force in Y-direction. (1.0)

CDRAGZ Multiplier [-] of drag force in Z-direction. (1.0)

```

+-----+
| Orphan Input |      The following variables exist in
+-----+ NAMELIST DATA.  They are associated with
                        obsolete or future model options.  VALUES
MUST NOT BE INPUT BY THE USER FOR THESE VARIABLES.

```

```

ALPHA      Time advancement of momentum and energy equations:
           0.0 .. Semi implicit (incomplete logic);
           1.0 .. Fully implicit (default).
AO  )  {    Constants in explicit convective flux calculations;
BO  )  {    range: 0.0 to 0.5; donor: AO=0.5, BO=0.0 (default);
        {    central: AO=0.0, BO=0.0.  Implicit is always donor.
DDDHEX     Select value of DDDH (change in DHDT with respect to
           enthalpy, where DHDT is change in enthalpy with
           respect to time) in energy equation solution; EXPLICIT
           mode only; may be:
           0.0..Use cell-by-cell values of DDDH. (default)
           <0.0..Use maximum cell value of DDDH for all cells.
           >0.0..Use DDDHEX for DDDH for all cells.
DLCUT      Multiplier [-] of 0.5*DCONV used to eliminate cells
           from the mass-momentum iteration for ISYMCH=2. (0.5)
           Also, old turbulence convergence parameter.
DPDX       ?..Old way of initializing pressure & flows.
DPDY       ?..Old way of initializing pressure & flows.
DPDZ       ?..Old way of initializing pressure & flows.
DTENER     Time step size [s] for EXPLICIT-mode energy calc.
DTWALL     Time step size [s] for duct wall calc. (internal)
HYDALL     Hydraulic diameter [m], overall, for Asy. (internal)
HYDOUT     Hydraulic diameter [m], outer, for Asy. (internal)
HYDRAD     ?..Hydraulic diameter; not used.
IDDDP      Select role of IDDDP (change in mass residual with
           respect to pressure) in pressure equation solution;
           EXPLICIT mode only; may be:
           0..Compute for each cell at every time step, then use
              maximum of these for all cells; useful for IGEOM>0
              when all cells are geometrically similar.
           1..Compute for each cell at ISTATE=0 only; use these
              values for entire case. (default)
           2..Compute for each cell at every time step; useful
              when large pressure drops are present.
ITENMX     Old max # iterations. (internal; always 1)
ITMASX     Old max # iterations. (internal; always 1)
ITMOMX     Old max # iterations. (internal; always 1)
NREBP      Future: # of Rebalance passes. (internal; always 1)
NREBRS(5)  Future: multi-pass Rebalance. (not used)
OMEGAR     <0 allows constant density calc.
XKS        Used as a local variable: solid conductivity. (internal)

```

```

+-----+
| End of NAMELIST DATA Input |
+-----+

```

* NAMELIST INPUTQ * This namelist is read only when IFROD>0
 ***** at ISTATE=0 or ISTATE=2(limited change).

IQ Type of axial power distribution (array QK); may be:
 0..user specified; input array QK. (default)
 1..cosine shape with peak at mid-height.
 2..cosine shape with peak skewed toward KHHS.
 3..cosine shape with peak skewed toward KLHS.

QK(K) Relative power [-] in axial node K. (Default is 1.0 for
 KLHS<=K<=KHHS and 0.0 for all other K.) Restrictions: Must
 have QK(K)>0 for KLHS<=K<=KHHS; values input will be
 normalized by the program. (CAUTION: Array QK is heat
 source multiplier for Thermal Structures, duct walls,
 QFLUX, and QSCOOOL.)

KLHS Lowest heated K-plane. (0)

KHHS Highest heated K-plane. (0) Restriction: IQ>0: KHHS<=49.

FNZ Ratio [-] of maximum-to-average axial power density
 for IQ>0. (0.0)

QFLUX Average heat flux [W/m²] at fuel pin surface. (0.0)
 Restriction: ISTRUC=0.

QSCOOOL Average direct heat source per coolant volume [W/m³] in
 addition to that described by Thermal Structures. (0.0)
 Restriction: ISTRUC>0. (If ISTRUC=0, this type of heat
 source, if used at all, must be described using 'QSOU'
 records in the Internal Cell Initialization input block.)

NOFQT Number of the transient function which is used as a
 multiplier of the heat source for the coolant when thermal
 structures are present and as a multiplier of total heat
 source when no thermal structures are present. (0)

QIN(IND) Relative power [-] for cell position (I,J), where
 QIJ(I,J)=QIN(I+IMAX*(J-1)). (0.0) Restrictions: IMAX<=10,
 JMAX<=10. (CAUTION: Array QIJ is heat-source multiplier
 for duct walls, QFLUX, and QSCOOOL.)

When IGEOM>0, the remaining variables in NAMELIST INPUTQ are set internally by the program; these internally set values should not usually be overridden for standard fuel assemblies. For nonstandard situations (e.g. IGEOM<1), these variables may be used to provide values for certain pin models.

CLADOD Clad outside diameter [m]. (Default set by value read in NAMELIST GEOM.) Used for QFLUX!=0 and IDRAG>0.

PITCHX Pitch in the X-direction [m]. (Defaults: DX(2) when IPART=1 and 2.0*DX(2) when IPART=0.) Used for IDRAG>0.

PITCHY Pitch in the Y-direction [m]. (Defaults: DY(2) when IPART=1 and 2.0*DY(2) when IPART=0.) Used for IDRAG>0.

IJTYPE(IND) Cell type. (0) Cell types are positive integers and are used as indices IJ of the following two variables. If a positive value is input for any of the following two variables, then the corresponding parameter will be set to that value in all cells of that type. The index IND is related to the cell location indices (I,J) by the following: $IND = I + IMAX * (J - 1)$. Restrictions: $IMAX \leq 10$, $JMAX \leq 10$, $IJTYPE(IND) \leq 50$.

PINAF(IJ) Fuel pin fraction [-] in cell type IJ, where $1 \leq IJ \leq \max(IJTYPE(IND))$. (0.0) Used to fill array PINF(I,J) when QFLUX!=0. CAUTION: If input any PINAF>0, then must input PINAF for all cell types.

WETLN(IJ) Wetted perimeter [m] in cell type IJ, where $1 \leq IJ \leq \max(IJTYPE(IND))$. (-1.0) Used to fill array WETLEN(I,J) when IDRAG>0. (In order to adjust the axial flow area, use 'ALZ ' and/or 'AL ' records in the Internal Cell Initialization input block.)

An example might help to clarify the input for the three variables above. Consider a case with $IMAX = JMAX = 6$.

ITYPE=12*1,24*2, indicates that cells (1,1) through (6,2) are assigned type 1 and cells (1,3) through (6,6) are assigned type 2.

PINAF=0.5,0.25, Cells of type 1 and 2 are given pin fraction values of 0.5 and 0.25, respectively.

WETLN(2)=0.28, Cells of type 2 are assigned wetted perimeter values of 0.28 m and cells of type 1 retain their default values (as set by the hexagonal geometry model if IGEOM>0).

```

+-----+
| Orphan Input |      The following variables exist in
+-----+      NAMELIST INPUTQ.  They are associated
                  with obsolete or future model options.
VALUES MUST NOT BE INPUT BY THE USER FOR THESE VARIABLES.

```

```

DTFUEL      Time step [s] for T.S. calculation. (internal)
GAMAHT      ?..Gamma heat contrib to power; not used.
HYDALL      Hydraulic diameter [m], overall, for Asy. (internal)
HYDIN       Hydraulic diameter [m], interior, for Asy. (internal)
HYDOUT      Hydraulic diameter [m], outer, for Asy. (internal)
IFMAX       ?..Not used.  (QGENER: 25)
NOFQZ       ?..Not used.
PINPOW(1)   Old way of setting relative pin powers.
QTOTAL      ?..Total bundle power; not used.

```

```

+-----+
| End of NAMELIST INPUTQ Input |
+-----+

```

 * REBALANCING REGION CARDS *

This set of cards must be present
 only when IFREB>0 in NAMELIST GEOM
 and NEWREB=1 in NAMELIST DATA;
 this implies ISTATE=0 or =2.

When present, this block is terminated by a card containing
 'END ' in columns 1-4. Rebalancing is not recommended
 when the conjugate gradient solution (ISYMCH=5) is used.

These cards are used to specify the location of rebalancing
 regions and rebalancing surfaces. Additional input is
 required (see Rebalancing Option section of NAMELIST DATA).
 Each card in this section contains the following variables
 in the FORMAT (A4,7I4):

NAME NUM IB IE JB JE KB KE

NAME REBM..The cells defined on this card form (part of) a
 rebalancing region. At least one card of this type
 is required for each region.
 REBX..The cells defined on this card define a rebalancing
 surface which coincides with an I grid plane. The
 surface defined by cell (I,J,K) is the surface between
 cell (I,J,K) and cell (I+1,J,K).
 REBY..The cells defined on this card define a rebalancing
 surface which coincides with a J grid plane. The
 surface defined by cell (I,J,K) is the surface between
 cell (I,J,K) and cell (I,J+1,K).
 REBZ..The cells defined on this card define a rebalancing
 surface which coincides with a K grid plane. The
 surface defined by cell (I,J,K) is the surface between
 cell (I,J,K) and cell (I,J,K+1).
 END ..This card terminates the REBALANCING REGION CARDS.

NUM Rebalancing region number.

IB,IE These six variables are the beginning and ending
 JB,JE I-, J-, and K-indices used to define a rectangular
 KB,KE solid of cells which constitute (part of) a
 rebalancing region or a plane of cells adjacent to
 a rebalancing surface.
 Note 1. Internal surfaces having zero-permeabilities
 should not be included as rebalancing surfaces.
 Note 2. The number of cells and surfaces input in these
 cards must exactly match the numbers specified
 in the variables NREBRT, NREBM, NREBX, NREBY, and
 NREBZ in the Rebalancing Option section of
 NAMELIST DATA.

```

*****
* FLUID CELL LOCATION CARDS *
*****

```

This set of cards must be included when IFHTX>1 in NAMELIST GEOM and ISTATE=0 in NAMELIST DATA.

This input block is terminated by a card with 'END ' in cols. 1-4.

These cards specify the fluid type which occupies each internal cell. ONLY those cells which do NOT contain Fluid #1 are to be specified. (Surface/Fluid information is also needed, as provided by KPROP in NAMELIST DATA.) The present input consists of the following variables in the FORMAT (A4,7I4):

```

NAME    NUM    IB    IE    JB    JE    KB    KE

```

NAME HTEX...Defines a region of cells containing Fluid # NUM.
 END ..This card terminates the alternate fluid location input; it is only present if IFHTX>1 and ISTATE=0.

NUM Fluid # for the cells specified: Enter a number between 2 and IFHTX, inclusive, to specify which alternate fluid is present.
 (Since the default for all cells is Fluid # 1, you may not enter NUM=1 in this block.)
 These Fluid #'s are stored in array INDHTX.

IB,IE These six variables are the beginning and ending I,J,and K indices of a rectangular(cylindrical) volume of the calculational domain which has Fluid # NUM present.
 JB,JE (These indices have the same meaning as for INTERNAL CELL INITIALIZATION CARDS.) A cell may only contain one fluid. Cells containing different fluids must be physically separated from each other using zero surface permeabilities (ALX,ALY,ALZ) or non-flow Boundary Surfaces.
 KB,KE

* NAMELIST PUMP * This input block is required when IFPUMP>0
***** and ISTATE=0 or 2 in NAMELIST DATA.

The information required depends on the pump model type, as selected by IFPUMP; all pumps use the same model type. Where indicated, input values for each pump, NP=1,2,...,NPUMP.

```
+-----+
| IFPUMP=1: Specified Speed Model (EBR-II functional form) |
+-----+
  DPPUMP = a*(NBAR**2 + b*NBAR*QBAR + c*|QBAR|**d)
  with
    NBAR = normalized pump speed (N/N_R)
    QBAR = normalized volume flowrate (Q/Q_R)
    a,b,c: user-supplied coefficients
    d: user-supplied laminar/turbulent exponent

NPUMP      Number of pumps; may not exceed 5.
PUMPS(NP)  N: Actual speed [revolution/minute] for pump NP. (0.0)
PUMPSR(NP) N_R: Reference speed [revolution/minute] for pump NP. (0.0)
NTPUMP(NP) nf: Transient function number corresponding to the
              normalized speed NBAR for pump NP. Must input
              1<=NTPUMP<=25 for ISTATE>=2.
QFLOWR(NP) Q_R: Reference volumetric flow rate [m**3/s] for pump
              NP. (0.0) (Sign has same convention as velocity.)
              Ignored for ISTATE<2; code uses QBAR=1.0.
              * BUG: Code currently uses QBAR=1.0 for all ISTATE. *
PUMLEN(NP) L: Characteristic length [m] in Reynolds number, Re,
              used in choosing between d_l and d_t for pump NP.
ICORP(NP)  Correlation flag for pump NP. (1) (MUST take default.)
NCORRP     Number of pump correlations. (1) (MUST take default.)

      Input the following 6 items for 1<=NCP<=NCORRP.
ACORP(NCP) a: Pressure equation coefficient [Pa].
BCORP(NCP) b: Pressure equation coefficient [-].
CCORP(NCP) c: Pressure equation coefficient [-].
ALPHAL(NCP) d_l: Pressure equation exponent [-] for Re<=Re_{tr}.
ALPHAT(NCP) d_t: Pressure equation exponent [-] for Re>Re_{tr}.
REYTRP(NCP) Re_{tr}: Reynolds number [-] for transition from
              exponent d_l (laminar) to d_t (turbulent).

IPUBUG     Normal and debug edit detail flag; see IFPUMP=3&4.
```

```
+-----+
| IFPUMP=2: Specified Pressure Increase Model (ISTATE>=2 only) |
+-----+
  DPPUMP = (Steady-State Value) * (Transient Function # nf)
  where Steady-State Value was computed by code for ISTATE<2
  using IFPUMP=1, 3, or 4.

NTPUMP(NP) nf: Transient function number corresponding to the
              normalized pressure increase for pump NP. Must input
              1<=NTPUMP<=25 for ISTATE>=2.
IPUBUG     Normal and debug edit detail flag; see IFPUMP=3&4.
```

```

+-----+
| IFPUMP=3 or 4: Homologous Model (Sec. 8.2, Ref. 6) |
|   IFPUMP=3: Full four-quadrant pump operation.   |
|   IFPUMP=4: Reverse pump rotation is precluded.   |
+-----+

```

NPUMP Number of pumps; may not exceed 5.

PUMPHR(NP) H_R: Reference head [m] for pump NP.

PUMPS(NP) N: Actual speed [revolution/minute] for pump NP. (0.0)

PUMPSR(NP) N_R: Reference speed [revolution/minute] for pump NP. (0.0)

QFLOWR(NP) Q_R: Reference volumetric flowrate [m**3/s] for pump
NP. (0.0) (Sign has same convention as velocity.)

TORQR(NP) tau_hR: Reference hydraulic torque [N*m] for pump NP.

TORQFR(NP) tau_fR: Reference friction torque [N*m] for pump NP
if ICORP(NP)=50.

ICORP(NP) Characteristic curve type for pump NP; may be:
 = 1: user-input characteristic curves (default);
 =50: built-in curves for CRBR primary sodium pumps.

When ICORP=1: Head and hydraulic torque are represented by eighth-order polynomials in $X (= \pi + \text{atan}(\text{QBAR}/\text{NBAR}))$; the default values (Ref. 8) shown below are for a specific speed ($N*Q^{.5}/H^{.75}$) of 35 [COMMIX units]. Frictional torque is represented by a three-range set of third-order polynomials in NBAR; the default values shown below are from Ref. 9, except for a low-speed modification to avoid numerical instabilities.

CORPH8(i) C_{h,i}: Coefficient i [-], $1 \leq i \leq 9$, in head function
WH(X). (0.698652605330, -0.849417704720, 2.21271612811,
-2.93805342603, 2.02577962042, -0.715158600838,
0.130660605500, -0.0117638212829, 0.000412183270823)

CORPT8(i) C_{\tau,i}: Coefficient i [-], $1 \leq i \leq 9$, in hydraulic
torque function WT(X) for IFPUMP=3. (-0.600209192472,
1.39811421649, -1.55466906832, 3.26133243601,
-3.06969387345, 1.34928570234, -0.301384097624,
0.0332164243168, -0.00143383177999)

CORPT8(i+9) C_{\tau,i+9}: Coefficient i [-], $1 \leq i \leq 9$, in hydraulic
torque function WT(X) for IFPUMP=4. (48.89663120,
-124.1188706, 121.4496110, -52.19548958, 5.140216254,
3.780294746, -1.506546075, 0.2197097906, -0.01175690460)

CORPF(i+3j) C_{f,i+3j}: Coefficient i [-], $1 \leq i \leq 3$, in frictional
torque function WF(NBAR) range j, $0 \leq j \leq 2$. (0.07215,
0.0, 0.0, 0.117, -8.97, 0.0, 0.012, 0.023, 0.0)

CUTPF(j+1) C_{f,cut,j+1}: Relative speed [-] at which to switch
from range j to j+1, $0 \leq j \leq 1$, in frictional torque
function WF(NBAR). (0.005, 0.0117)

When ICORP=50: Head and both torque functions are in the code based on Ref. 10; the user may only override the built-in hydraulic torque coefficients:

CORPT(i) C_{\tau,i}: Coefficient i [-], $2 \leq i \leq 6$, in hydraulic
torque function WT(NBAR/QBAR). (0.6500, 0.2738,
0.2512, -0.1325, 0.0051) Note: CORPT(1) is not used.

(Continuation of input for IFPUMP=3 or 4)

DENRO rho*g: Product of fluid density and gravitational
acceleration at reference conditions [kg/m**2/s**2].
IPUBUG Normal and debug edit detail flag; may be:
 =0: no output (default);
 =1: 1-line summary;
 >=2: 3-line summary (QFSR=Q/Q_R, PSSR=N/N_R,
 DPPUMP=pump pressure increase, etc.);
 >=3: transient function (debug);
 >=4: torque and head (debug);
 >=5: transient function coefficients (debug).

The remaining input variables are only used for ISTATE>=2.

Restrictions on next 2 items: input either NTORQM(NP)>0
(typical) or NTPUMP(NP)>0; max. value = 25.
NTORQM(NP) nf: Transient function number corresponding to
 normalized motor torque for pump NP; speed will be
 calculated dynamically by code.
NTPUMP(NP) nf: Transient function number corresponding to
 normalized speed for pump NP.

PINER(NP) I_p: Moment of inertia [kg*m**2] for pump NP if
 NTORQM(NP)>0.
TORQM(NP) tau_mR: Reference motor torque [N*m] for pump NP if
 NTORQM(NP)>0. (Default: based on steady-state result.)
IPTRAN Motor torque integration method for NTORQM>0; may be:
 =0: approximate;
 =1: exact (default).

The user may force use of a linear head versus flow rate
relationship at low flow rate. At the first time that
QBAR < Q_{cut} and NBAR < N_{cut} ,
for any pump having ICORP(NP)=1, the code will set
HTEST = current HBAR and QTEST = current QBAR ;
thereafter, the relative head will be calculated from
HBAR = (HTEST/QTEST) * QBAR
for all pumps having ICORP(NP)=1. User inputs the following
two variables; input QCUT<<0.0 and NCUT<<0.0 to bypass use
of this linear head versus flow rate relationship.
QCUT Q_{cut}: Relative flow rate [-] below which a linear
 head vs. flow rate relationship is used. (0.01)
NCUT N_{cut}: Relative speed [-] below which a linear head
 vs. flow rate relationship is used. (0.001)
Note: NCUT is a REAL, not INTEGER, variable.

```
+-----+
| Orphan Input |   The following variables exist in
+-----+ NAMELIST PUMP. They are associated with
              obsolete or future model options.  VALUES
MUST NOT BE INPUT BY THE USER FOR THESE VARIABLES.
```

```
CORPH(15)    ?..Not used.
CUTPT(3)     ?..Not used.
PUMPH(NP)    Head [m]; set internally for IFPUMP=3&4.
QCUTC        HTEST/QTEST; set internally for IFPUMP=3&4.
TORQ(NP)     Hydraulic torque [N*m]; set internally for IFPUMP=3&4.
```

```
+-----+
| End of NAMELIST PUMP Input |
+-----+
```

 * PUMP LOCATION CARDS *

This set of cards is required when IFPUMP > 0 in NAMELIST DATA both when ISTATE = 0 and when ISTATE = 2. This input block is terminated by a card containing 'END' in col. 1-4.

These input are used to locate the pump(s) described in NAMELIST PUMP. Pumps should be placed in single cells only which are completely blocked to flow in the four directions normal to the fluid flow through the pump. (See example below.) The pressure increase/decrease from the pump is placed on the cell surface in the increasing nodalization direction along the direction of fluid flow; i.e., in the case of flow in the Z-direction (either up or down), the pressure change is placed between the K and K+1 cells when the pump is located in the K-th cell as shown below.

Each card in this section contains the following variables in the FORMAT (A4,7I4):

NAME	NUM	I	I	J	J	K	K
NAME Pump direction indicator:							
	PUMX..X(R)-direction fluid motion.						
	PUMY..Y(THETA)-direction fluid motion.						
	PUMZ..Z-direction fluid motion.						
	END ..This card terminates the PUMP LOCATION CARD input.						
NUM	Pump number (<=NPUMP).						
I	These variables are the I-, J-, and K-indices used to define a face of a cell. The cell face defined by cell (I,J,K) for X(R)-direction flow is the face between cells (I,J,K) and (I+1,J,K); for Y(THETA)-direction flow, it is the face between cells (I,J,K) and (I,J+1,K); for Z-direction flow, it is the face between cells (I,J,K) and (I,J,K+1). Only a single cell may be specified for each pump.						
J							
K							

Example: If pump # 2 supplies a Z-direction pressure increase at the interface between cells (3,5,7) and (3,5,8), then enter the PUMP LOCATION CARD:

PUMZ 2 3 3 5 5 7 7

and enter INTERNAL CELL INITIALIZATION CARDS:

ALX 0.0 2 3 5 5 7 8

ALY 0.0 3 3 4 5 7 8

(or otherwise restrict the flow path using Boundary Surfaces).

```
*****
* FORCE STRUCTURE LOCATION CARDS *
*****
```

This set of cards must be included only when NFORCE > 0 in NAMELIST GEOM and NEWFOR = 1 in NAMELIST DATA. This input block is terminated by a card containing 'END' in col. 1-4.

These cards are used to locate the force structures described in the Force Structure section of NAMELIST DATA. These forces can be applied at cell faces between two computational cells. The locations therefore correspond to portions of grid planes. Each card in this section contains the following variables in the FORMAT (A4,7I4).

```
NAME  NUM  IB  IE  JB  JE  KB  KE
```

NAME Force direction indicator:
 XFOR..X(R)-direction force.
 YFOR..Y(THETA)-direction force.
 ZFOR..Z-direction force.
 END ..This card terminates the FORCE STRUCTURE LOCATION CARDS.

NUM Force structure number (<=NFORCE).

IB,IE These six variables are the beginning and ending
 JB,JE I-, J-, and K-indices used to define a plane of
 KB,KE cells. The cell face defined by cell (I,J,K) for
 a X(R)-direction force is the face between cells
 (I,J,K) and (I+1,J,K); for a Y(THETA)-direction
 force, it is the face between cells (I,J,K) and
 (I,J+1,K); for a Z-direction force, it is the face
 between cells (I,J,K) and (I,J,K+1). A specific
 cell (I,J,K) may have a different force specified
 for each of the three associated faces; however,
 there may be only a single force specified for
 each cell face.


```
*****
* NAMELIST STRUCT *      This set of cards is included if and
*****                  only if ISTRUC = 2 in NAMELIST GEOM
and NEWTS = 1 in NAMELIST DATA. This implies that ISTATE=0 or 2.
```

```
ITSBUG      0--No thermal structure debugging. (default)
            1--Some thermal structure debugging is printed.
              (Primarily T.S. Control Block with index offsets.)
```

```
*****
* THERMAL STRUCTURE PROTOTYPE CARDS *  This set of cards is
*****                                included when ISTRUC=2 in
                                  NAMELIST GEOM and NEWTS=1
in NAMELIST DATA; this implies that ISTATE=0 or 2. If a prototype
is to be changed at ISTATE=2, then input must be present for ALL
prototypes and locations, not just for the ones being changed; the
preexisting temperature distributions in all T.S. are lost.
A card containing 'END ' in cols. 1-4 must follow the final prototype
to terminate this input block. The maximum number of prototypes
allowed is set by PARAMETER PNSTRU in the source code; the
current value of PNSTRU is 99.
```

A thermal structure is a collection of thermal structure elements each of which has the same characteristics as specified by a thermal structure prototype. Thermal structure prototypes are defined using NAMELISTs T, F, and M standing for Type, Fluid, and Material, respectively. The order in which these NAMELISTs are input indicates the construction of the thermal structures and must conform to the following rules:

1. A NAMELIST T must begin the definition of each thermal structure prototype; there is only one per prototype.
2. One, or more, NAMELISTs M follow NAMELIST T. The ordering of these cards is from OUT-side surface to IN-side surface and determines the composition.
3. A gap exists after each material except the last. The gap parameters are specified in the NAMELIST M.
4. A NAMELIST F preceeds NAMELISTs M if fluid interacts with the OUT-side surface and follows the NAMELISTs M if fluid interacts with the IN-side surface. At least one NAMELIST F must be present; both may be present.
5. The initial default value set for most variables is not valid. Subsequent defaults are the values in effect after reading the previous namelist. If, for example, the geometrical type is the same for all thermal structure prototypes, IXYZ need be specified only on the first NAMELIST T. (Exception: KRAD=0 set before reading each NAMELIST M.)
6. The definition of thermal structure prototype N+1 must follow the definition of thermal structure prototype N.
7. Blank cards or cards with blanks in columns 1 through 4 may be interspersed as desired between namelists.

The precise definition of each namelist is as follows:

```

+-----+
| NAMELIST T - Type |
+-----+

```

N Thermal structure prototype number. This number does not need to correspond to its index or ordinal number. Remember, the thermal structures must be input in ascending order: T.S. N+1 must follow T.S. N. The maximum value for N is PNSTRU (above).

IXYZ Geometrical type or characteristic.

- 1..Rods (cylinders) with axis aligned in the I-direction.
- 2..Rods (cylinders) with axis aligned in the J-direction.
- 3..Rods (cylinders) with axis aligned in the K-direction.
- 11..Slab with the normal aligned in the I-direction.
- 12..Slab with the normal aligned in the J-direction.
- 13..Slab with the normal aligned in the K-direction.
- 101..Sphere aligned in the I-direction.
- 102..Sphere aligned in the J-direction.
- 103..Sphere aligned in the K-direction.

(The alignment specification determines which one of the indices I, J, or K is used as the index m in the "axial" power distribution multiplier, QK(m).)

NT The number of the transient function to be used as a multiplier for the heat source. (These values will be overridden by any input for NTOTS in NAMELIST DATA.)

RODFR Thermal structure multiplicity factor; input meaning depends on geometry (cf. Sec. 5.2.1 of User's Guide):

Rods or cylindrical thermal structures [-]:

- >0..Number or fraction of actual rods interacting with each associated coolant cell.
- <0..The absolute value is the number or fraction of rods per unit area (m^2 , unaffected by cell porosities) interacting with each associated coolant cell. (For a two-sided T.S.: the reference area of the associated cell is that cell which interacts with the OUT-side surface of the T.S.) The unit area has the same normal as the axis of the cylinder.

Slab thermal structures:

- >0..Slab area [m^2] in each associated coolant cell.
- <0..The absolute value [-] is the slab area divided by the cell area (unaffected by cell porosities). In the case of two sided thermal structures, this value is equivalent to a surface porosity for the structure. The slab and cell area have the same normal.

Spherical thermal structures [-]:

- >0..Number or fraction of spheres interacting with each associated coolant cell.
- <0..The absolute value is the number or fraction of spheres per unit volume (m^3 , unaffected by cell porosities) interacting with each associated coolant cell.

OUTR Thermal structure outer radius [m]. (This is not used slab type thermal structures.)

```

+-----+
| NAMELIST F - Fluid |
+-----+

```

IHT Heat transfer correlation index. This value is used as the index NH of variables HEATC1, HEATC2, HEATC3, and HEATC4 or the index nh in user-supplied function HEATCU as described in the Fluid-Solid Heat Transfer section of NAMELIST DATA.

HYD Hydraulic diameter or reference length [m]. This value is used as D, the reference length, as described in the Fluid-Solid Heat Transfer Section of NAMELIST DATA.

```

+-----+
| NAMELIST M - Material |
+-----+

```

MI Material type index. This value is used as the index NM described in the Material Properties (Solids) Section of NAMELIST DATA.

NP Number of partitions in the material. A thermal structure temperature will be computed for each material partition. The sum of NP over all NAMELISTs M for a single Thermal Srtucture Prototype must be ≤ 49 .

DR Partition size [m]. (Note: total thickness = NP * DR.)

Q Volumetric heat source [W/m³] for the material region. Q will be multiplied by QK(I or J or K, depending on IXYZ) and by the NT-th transient function during the calculation.

The following gap properties must be correctly specified or defaulted when another NAMELIST M follows the current NAMELIST M; the gap properties on the final NAMELIST M in a prototype are ignored.

SGAP Gap size [m]. Only used for cylinder and sphere shapes.

HGAP Gap heat transfer coefficient [W/m²/C]. Need HGAP>0.

KRAD Radiating Surface Number. Only used when IFRAD > 0; relates to FVIEW, EMITS; and NTF for KFLOW=200+NTF. Both the OUT- and IN-sides of a T.S. may radiate. If the OUT-side surface of a T.S. radiates, specify non-zero KRAD on the first NAMELIST M in the prototype. If the IN-side surface of a T.S. radiates, there must be at least two NAMELISTs M and specify non-zero KRAD on the last NAMELIST M in the prototype. Non-zero KRAD values are ignored in all other NAMELISTs M. The proper surface's KRAD must be the same as NTF if using flow boundary condition of the form KFLOW=200+NTF.

-- See example on page 57. --


```

+-----+
| Example of T.S. Input for IFRAD>0 |
+-----+

```

The following example illustrates the use of KRAD in the T.S. prototype definitions. Prototype N=3 is two-sided and the OUT side is radiating surface (KRAD) #1. Prototype N=4 is two-sided and both surfaces are radiating; the OUT side is radiating surface (KRAD) #3 and the IN side is radiating surface #2. Prototype N=5 is one-sided and this (IN) side is radiating surface (KRAD) #4. The appropriate layout of the FVIEW array is shown: Rad.Surf.#1 radiates to Rad.Surf.#2 and Rad.Surf.#3 radiates to Rad.Surf.#4; since these are cylinders, Rad.Surfs.#2 and #4 'see' themselves. Radiation surface #1 also controls a flow B.C. via KFLOW=201.

&DATA

```

. . . (other input) . . .
KFLOW= . . . ,201, . . . ,
. . . (other input) . . .
IFRAD=2, NRAD=4, OMR=.2, EMITS=4*0.7,95*0.,
FVIEW=0. . .95 .0. .0. .95*0.,
      1. .05 .0. .0. .95*0.,
      0. .0. .0. .91 .95*0.,
      0. .0. .1. .09 .95*0., 9405*0.,
. . . (other input) . . .

```

&END

```

. . . (other input: e.g. INPUTQ,rebalance,HTEX,pump,forces) . . .

```

&STRUCT ITSBUG=0, &END

```

. . . (prototype definitions for N=1 and 2) . . .

```

```

&T N=3, IXYZ=03, RODFR=-.69, OUTR=2.87, NT=0, &END REACTOR VES
&F IHT=3, HYD=0.2540, &END
&M MI=2, NP=1, DR=0.0001, Q=0.0, KRAD=1, &END
&M MI=2, NP=1, DR=0.0253, Q=0.0, KRAD=0, &END
&F IHT=2, HYD=0.5080, &END

&T N=4, IXYZ=03, RODFR=-.21, OUTR=3.02, NT=0, &END GUARD VES
&F IHT=4, HYD=0.3670, &END
&M MI=2, NP=1, DR=0.0004, Q=0.0, KRAD=3, HGAP=1.E3, SGAP=0., &END
&M MI=2, NP=5, DR=0.0100, Q=0.0, KRAD=0, &END
&M MI=2, NP=1, DR=0.0004, Q=0.0, KRAD=2, &END
&F IHT=3, HYD=0.2540, &END

```

```

&T N=5, IXYZ=03, RODFR=-.27, OUTR=3.33, NT=0, &END COLLECT SRF
&M MI=2, NP=1, DR=0.0253, Q=0.0, KRAD=0, &END
&M MI=2, NP=1, DR=0.0001, Q=0.0, KRAD=4, &END
&F IHT=4, HYD=0.3670, &END

```

```

. . . (prototype definitions for N>5) . . .

```

END (terminates all T.S. prototype definitions)

```

. . . (location cards for N=1 and 2) . . .

```

```

IN 3 6 6 1 4 1 10 RV-IN
OUT 3 7 7 1 4 1 10 RV-OUT
IN 4 8 8 1 4 1 10 GV-IN
OUT 4 9 9 1 4 1 10 GV-OUT
IN 5 10 10 1 4 1 10 CS-IN

```

```

. . . (location cards for N>5) . . .

```

END (terminates all T.S. location specifications)

```

. . . (other input: boundary/cell initialization) . . .

```

```
*****
* BOUNDARY VALUE INITIALIZATION CARDS *      This set of card is
***** always present; even if
                                              no values are to be input,
the 'END ' card must be present (ISTATE=0/1/2/3).
```

The purpose of this set of input cards is to permit initialization of boundary values of any of the arrays listed below. Uniform temperature and velocity boundary conditions can be more easily specified using the variables TEMP and VELOC in NAMELIST DATA. Each card in this section contains the following variables in the FORMAT (A4,F10.3,7I4):

	NAME	RVAL	IB	IE	JB	JE	KB	KE	N
NAME	AREA..Surface element area [m ²]. (see BOUN.SURF.ID.CARD) HLB ..Enthalpy [J/kg]. (f(TLB,PB)) (See Note 1.) PB ..Pressure [Pa]. (PRES(N)) (See Note 1.) QBN ..Heat flux [W/m ²]. (TEMP(N), KTEMP=200,300+NTF) RLB ..Density [kg/m ³]. (f(PB,TLB)) (See Note 1.) TLB ..Temperature [C]. (TEMP(N), KTEMP<=200 & >=400) VELB..Magnitude of the velocity [m/s] normal to the surface in the direction indicated by XNORML(N), YNORML(N), and ZNORML(N). (VELOC(N)) END ..This card terminates the BOUNDARY VALUE INITIALIZATION CARDS. It must always be included, even if no boundary values were input in the run.								
RVAL	The value to be assigned to the variable named.								
IB,IE JB,JE KB,KE	These six variables are the beginning and ending I-, J-, and K-indices that define a rectangular solid composed of one or more cells. The rectangular solid that defines or partially defines a surface is the one which is totally interior and adjacent to, or partially interior to and intersecting that surface. Note. The scheme to indicate surfaces in the BOUNDARY VALUE INITIALIZATION CARDS is the same as that used to indicate surfaces in the BOUNDARY SURFACE IDENTIFICATION CARDS. (This, however, is different from the scheme used to indicate surfaces in the INTERNAL CELL INITIALIZATION CARDS.) Surface elements are indicated by the cell which is adjacent to and on the side pointed to by the surface normal.								
N	The surface number of the boundary being set.								

Note 1: There is no constructive purpose served by trying to override the defaults provided for HLB, PB, and RLB.

 * INTERNAL CELL INITIALIZATION CARDS *

This set of cards is
 always present; even if
 no values are to be input,

the 'END' card must be present (ISTATE=0/1/2/3).

The purpose of this set of input cards is to permit
 initialization of internal cell values of any of the arrays
 listed below. Each card of this section contains the
 following variables in the FORMAT (A4,F10.3,6I4):

NAME RVAL IB IE JB JE KB KE

NAME AL ..Volume porosity [-]: the ratio of fluid volume in a
 cell to total cell volume. (1.0) Need AL>0.
 ALX ..X(R)-direction surface porosity [-]: ratio of flow
 area to total area between cell (I,J,K) and cell
 (I+1,J,K). (1.0) Need ALX>=0.
 ALY ..Y(THETA)-direction surface porosity [-]: ratio of
 flow area to total area between cell (I,J,K) and
 cell (I,J+1,K). (1.0) Need ALY>=0.
 ALZ ..Z-direction surface porosity [-]: ratio of flow
 area to total area between cell (I,J,K) and cell
 (I,J,K+1). (1.0) Need ALZ>=0.
 { HL ..Enthalpy [J/kg]. (f(TL,PSTAT0)) Do NOT input HL! }
 P ..Pressure minus reference state pressure [Pa]. (0.0)
 QSOU..Volumetric heat source per computational cell volume
 (e.g. DX(I)*DY(J)*DZ(K)) [W/m^3]. (0.0) (Do not input
 if using Thermal Structures anywhere in model.)
 TL ..Temperature [C]. (TEMP0(NI))
 TURC..Turb. conductivity [W/m/C]. (TURBC) Need ITURKE=1.
 TURV..Turb. viscosity [Pa*s]. (TURBV) Need ITURKE=1.
 UL ..X(R)-component (on I+ face) of velocity [m/s]. (0.0)
 VL ..Y(THETA)-component (on J+ face) of velocity [m/s]. (0.0)
 WL ..Z-component (on K+ face) of velocity [m/s]. (0.0)
 END ..This card terminates the INTERNAL CELL INITIALIZATION
 CARDS. It must always be included, even if no
 internal cell values were input in the run.

RVAL The value to be assigned to the variable named.

IB,IE These six variables are the beginning and ending I-,
 JB,JE J-, and K-indices that define a rectangular(cylindrical)
 KB,KE solid composed of one or more cells.

Note. The scheme to indicate surfaces in the
 INTERNAL CELL INITIALIZATION CARDS is
 different from that used to identify surfaces
 in the BOUNDARY SURFACE IDENTIFICATION CARDS
 or BOUNDARY VALUE INITIALIZATION CARDS. Here
 cell (I,J,K) indicates the surface between
 that cell and either cell (I+1,J,K), (I,J+1,K),
 or (I,J,K+1), whichever is appropriate for the
 variable being initialized. Surfaces lying on
 boundaries must not be initialized using the
 INTERNAL CELL INITIALIZATION CARDS but rather
 using the BOUNDARY VALUE INITIALIZATION CARDS.

```
* * * * *
* * * * *
* *
* *      END OF COMMIX-1AR/P INPUT DESCRIPTION      * *
* *
* * * * *
* * * * *
```



```

* * * * *
* * * * *
* *
* *      Appendix A (to the COMMIX-1AR/P Input Description)
* *
* *
* *      BEHIND THE SCENES . . .
* *
* * * * *
* * * * *

```

```

*****
* CONTROL PARAMETERS AT A GLANCE *
*****

```

The two tables below are included to clarify the role of some of the control parameters. The values indicated, while not guaranteed, are ones that have been found to work in many applications. The user is encouraged to optimize these parameters according to application. A double asterisk indicates that the parameter is not used. A value enclosed in brackets '[]' indicates that the default value is different and that this value must be explicitly specified in the input. For example, the default for RDTIME is 0.8. When running with ISYMCH=3 however, the recommended value of RDTIME is 10.0. Therefore in NAMELIST DATA the specification 'RDTIME=10.0' must appear.

```

+-----+
| Implicit Time Advancement |
+-----+

+---Time Step Loop
| NTMAX          99999
| TIMAX          3.6E+7
| IDTIME         [0]      [1]      2
| TSTART         0.0
| DT(1)          0.1
| DT(2)          0.1
| LASTDT         99999
| RDTIME         **      [10.0]    **
| NTHCON         -1
|
|   +---Outer Iteration Loop
|   | IT(1)       500
|   | IT(2)       500
|   | LASTIT      99999
|   |
|   |   OMEGAV          1.0
|   |   +---Pressure Iteration Loop
|   |   | ISYMCH
|   |   | |
|   |   | +-----+-----+
|   |   | |         |         |
|   |   | | [3]     [4]     5
|   |   | | 99      99      99
|   |   | | OMEGA    1.5    [0.95]  **
|   |   | | EPS1     1.0E-9  1.0E-9  1.0E-9
|   |   | | EPS2     1.E-11  1.E-11  1.E-11
|   |   | +---End of Pressure Iteration Loop
|   |   | +---Turb. Kinetic Energy Iter. Loop (ITURKE>10)
|   |   | | ITMAXK      99
|   |   | | RELAXK      0.8
|   |   | | EPS6        1.0E-9
|   |   | +---End of Turbulent Kinetic Energy Iteration Loop
|   |   | +---Turb. Kinetic Energy Dissipation Iter. Loop (ITURKE>11)
|   |   | | ITMAXK      99
|   |   | | RELAXK      0.8
|   |   | | EPS6        1.0E-9
|   |   | +---End of Turbulent Kinetic Energy Dissipation Loop
|   |   | OMEGAE        0.8
|   |   | +---Energy Iteration Loop
|   |   | | IFITEN
|   |   | | |
|   |   | | 3
|   |   | | ITMAXE      99
|   |   | | RELAXE      0.95
|   |   | | EPS5        1.0E-9
|   |   | +---End of Energy Iteration Loop
|   |   EPS3          1.0E-6
|   | +---End of Outer Iteration Loop
|   +---End of Time Step Loop

```

```

+-----+
| Explicit Time Advancement |
+-----+

```

```

+--Time Step Loop

```

```

| NTMAX      99999
| TIMAX      3.6E+7
| IDTIME     [1]
| TSTART     0.0
| DT(1)      0.1
| DT(2)      0.1
| LASTDT     99999
| RDTIME     0.8
| NTHCON     -1

```

```

|   +--Mass-Momentum Iteration

```

```

|   |
|   |               ISYMCH
|   |               |
|   |   +-----+-----+
|   |   |       |       |
|   |   | [0]   | [1]   | [2]
|   |   +-----+-----+
|   | IT(1)     [10]   [10]   [10]
|   | IT(2)     [10]   [10]   [10]
|   | LASTIT    99999  99999  99999
|   | OMEGA     [0.95]  1.5    1.5
|   | IDDDP     1       1      1
|   | DLCUT     **      **     0.5
|   | EPS1      [1.0E-4][1.0E-4][1.0E-4]

```

```

|   +--End of Mass-Momentum Iteration

```

```

|   +--Energy Solution

```

```

|   |
|   |               IFITEN
|   |               |
|   |   +-----+-----+
|   |   |       |       |
|   |   | [0]   | [1]   | [2]
|   |   +-----+-----+
|   | DDDHMX    **      0.0    0.0

```

```

|   +--End of Energy Solution

```

```

+--End of Time Step Loop

```

```
*****
* STEADY-STATE DEFINITION *
*****
```

Calculationally, steady-state is declared by the code when the following conditions are met for all cells:

1. Fluid mass conservation satisfies: $\max(DL)/DCONV < 1$, where DL is the mass residual for a cell given by $DL = [d(\text{Density})/d(t) - \text{sum}(\text{Density} * \text{Area} * \text{Velocity})]/\text{Volume}$ and the normalization factor is (see SUBROUTINE GDCONV) $DCONV = EPS1 * (UVWMAX + EPS2)$, where $UVWMAX = \max[(\text{Density} * \text{Area} * \text{Velocity})/(\text{Volume})]$ and this maximization is over all faces of all cells.
2. The change of the U-velocity component divided by the maximum velocity magnitude in the entire field is less than EPS3.
3. The change of the V-velocity component divided by the maximum velocity magnitude in the entire field is less than EPS3.
4. The change of the W-velocity component divided by the maximum velocity magnitude in the entire field is less than EPS3.
5. The change in cell enthalpy divided by the current enthalpy is less than EPS3 (over 2 consecutive steps?).

EPSx input is described on page 22. It is the users responsibility to ensure that the solution has reached acceptable steady-state levels for the problem of interest with respect to other parameters which may still be slowly changing. In particular, temperature drifts in the thermal structures are not checked in the the above steady-state definition.

 * ERROR MESSAGES *

The following table gives a listing of the error messages processed by SUBROUTINE ERRCHK. It is intended that this section will be ever expanding thus making the running of COMMIX-1AR/P easier. In many cases, relevant information is printed out in the line(s) before the error message block. When appropriate, the error messages below refer to the variables in this information line. Variables are identified by their type (R=real, I=integer, and A=literal) and their relative position (1 through 12) in the line. For example, A1,R2,I3,I4,I5,I6,I7,I8,I9 would be used to refer to the variables printed in the following line:

ALX 1.0 3 4 2 8 1 9 6

IER	SUBROUTINE	ERROR DESCRIPTION
1	RESTAR	In reading the restart file, a block from COMMON /SPACE/ was found to have a length different from that calculated/expected by the code. Also used to indicate that the restart file was not completely read in preference to a new set of input cards, e.g. thermal structure override.
2	AMAIN	Boiling is starting to occur. (Norm.termination,print)
3	AMAIN	DTIME is less than or equal to zero. Check NAMELIST DATA array DT for user specified time step size or IDTIME and RDTIME for code determined time step size. In order to run with IDTIME=1 there must be a nonzero velocity somewhere on the boundary or in the interior. If none exists then one must set IDTIME=0 and specify a time step size in DT.
4	BARIN	An invalid input card has been encountered either in the BOUNDARY VALUE INITIALIZATION CARDS or the INTERNAL CELL INITIALIZATION CARDS. One of the following necessary conditions has not been satisfied: 1 <= IB <= IMAX, 1 <= IE <= IMAX, 1 <= JB <= JMAX, 1 <= JE <= JMAX, 1 <= KB <= KMAX, 1 <= KE <= KMAX, 1 <= N <= NSURF, IB <= IE, JB <= JE, KB <= KE, 0.0 < volume porosity (AL) , or 0.0 <= surface permeability (ALX, ALY, ALZ) <= 1.0. The card printed above the error message is ignored and processing continues. (Note: AL>1 will result in a warning message but the card is NOT ignored.)
5	BARIN	The surface indicated on a BOUNDARY VALUE INITIALIZATION CARD has no surface element or area.
6	BARIN	Invalid variable name on BOUNDARY VALUE INITIALIZATION CARD
7	BARIN	BOUNDARY VALUE INITIALIZATION CARD contains an INTERNAL CELL INITIALIZATION CARD variable name.
8	BARIN	Invalid variable name on INTERNAL CELL INITIALIZATION CARD
9	BARIN	INTERNAL CELL INITIALIZATION CARD contains a BOUNDARY VALUE INITIALIZATION CARD variable name.

- 10 AAMAIN The DIMENSION of variable S in COMMON /SPACE/ in SUBROUTINE ALTER is too small for the input values for this run. Change this DIMENSION to the value indicated in the printout; recompile, relink, and rerun.
- 11 BOXES One of the indices of the above BOUNDARY SURFACE IDENTIFICATION CARD is outside of one of the following ranges:
 I=1,IMAX J=1,JMAX K=1,KMAX N=1,NSURF,
 or one or more of the beginning indices is greater than the corresponding ending index, i.e.
 IB > IE or JB > JE or KB > KE.
- 12 BOXES While processing the card printed above the error box, a surface element was found to be specified as being contained in two surfaces. The cell and surface identifiers are (I1,I2,I3) and I4 and I5 on the second line.
- 13 BOXES On BOUNDARY SURFACE IDENTIFICATION CARDS, surfaces must be specified so that surface numbers are in increasing sequential order with all irregular surfaces preceding regular surfaces.
- 14 ERRCHK Only fifteen calls to ERRCHK are allowed before termination. This number can be increased by changing the value of NCALLS in SUBROUTINE ERRCHK.
- 15 FILLM One of the indices I, J, or K is outside of its expected range 1-IMAX, 1-JMAX, or 1-KMAX, respectively. This error usually occurs when the BOUNDARY SURFACE IDENTIFICATION CARDS have left a hole in the boundary. Recheck the BOUNDARY SURFACE IDENTIFICATION CARDS for an undefined or incorrectly defined surface and see the Appendix section entitled FINDING HOLES IN THE BOUNDARY.
- 16 GEOM3D Execution is terminating after processing geometry due to previously noted errors or LMPRNT>0.
- 17 FILLM The total number of cells counted in FILLM has exceeded the upper bound of IMAX*JMAX*KMAX. Recheck the BOUNDARY SURFACE IDENTIFICATION CARDS.
- 18 FILLM Excessive wrap around in the THETA direction. Check BOUNDARY SURFACE IDENTIFICATION CARDS; assume that all surface normals are pointing into the calculational area; if modeling full 2pi, sum of DY(j) must be within 0.0001 radian of 2pi.
- 19 FILLM Time has run out(<10s-CPU) while attempting to number the cells in FILLM. Probably caused by an input error in the BOUNDARY SURFACE IDENTIFICATION CARDS.
- 20 BOXES The number of surface elements has exceeded the value of NL1 as specified in NAMELIST GEOM. If the input value is correct check the BOUNDARY SURFACE IDENTIFICATION CARDS for possible errors.
- 21 BOXES The number of cells has exceeded the value of NM1 as specified in NAMELIST GEOM. If the input value is correct check the BOUNDARY SURFACE IDENTIFICATION CARDS for possible errors.
- 22 ----- -- Not currently used. --
- 23 AAMAIN Changes in NM1, NL1, IMAX, JMAX, and KMAX are not allowed when restarting (IFRES=2 or IFRES=3).

- 24 INITA3 INHTX When using the Simplified Properties or Multiple Fluids Options, you must input non-zero values for FC0H(i), FC1H(i), FC0RO(i), FC0K(i), and FC0MU(i) for each fluid(i) being used. FC0RO(i) and GASMW(i) may not both be input as non-zero.
- 25 INITIAL All surfaces, N, with a transient duct wall temperature boundary condition (KTEMP(N)=500+NTF) must have positive values input for: the material number, MATWAL(N); the fluid-structure heat transfer correlation number, IHTWAL(N); the duct wall thickness, WALLDX(N); and the characteristic length, HYDWAL(N). N, MATWAL(N), IHTWAL(N), WALLDX(N), and HYDWAL(N) are printed above the error message as I1, I2, I3, R4, and R5.
- 26 INITIAL A zero value of TEMP0(NI) has been found in INITIAL. Input a non-zero value in NAMELIST DATA and rerun.
- 27 INFORC An invalid input card has been encountered in SUBROUTINE INFORC while reading the FORCE STRUCTURE LOCATION CARDS. The first field must contain either 'XFOR', 'YFOR', 'ZFOR', or 'END '. The indices IB and IE, JB and JE, and KB and KE must refer to cells containing fluid. On the card printed above, cells after the the first invalid cell have been ignored and execution continues. A warning is issued if a later specification overrides a previous specification for a cell face.
- 28 IREBAL An invalid input card has been encountered in the REBALANCING REGION CARDS. One of the following errors has been sensed: Strings other than 'REBM', 'REBX', 'REBY', 'REBZ', or 'END ' in columns 1 through 4; One of the indices N, IB, IE, JB, JE, KB, or KE is out of its appropriate range (1,NREBRT), (1,IMAX), (1,JMAX), (1,KMAX); IB > IE, JB > JE, or KB > KE. The card printed above the error message is ignored and processing continues.
- 29 IREBAL A cell number (M) could not be found for the cell/face with indices (I,J,K), where I=I9, J=I10, and K=I11 from the line printed above the error message. Processing continues with the next cell/face.
- 30 IREBAL More cells have been found in the rebalancing region (surface) specified in the REBALANCING REGION CARDS than were specified by NREBM (NREBX, NREBY, or NREBZ) in the Rebalancing Option section of NAMELIST DATA. Execution terminates.
- 31 IREBAL The rebalancing regions and surfaces as specified by the variables of the Rebalancing Option section of NAMELIST DATA are inconsistent with the regions and surfaces as specified by the REBALANCING REGION CARDS. A comparison of the totals can be found in the tables above the error message. Execution terminates.
- 32 INITIAL ISTATE has been found to be 0 while attempting to restart from a previous run. It has been reset by the code to 1. Verify that this is an acceptable fix. Execution continues.
- 33 IREBAL The value of IFREB must be at least as large as the sum of NREBM(N), NREBX(N), NREBY(N) and NREBZ(N) over all regions N. The input value and minimum acceptable value are printed in the Rebalancing Summary above the error message. Reset IFREB and rerun.

- 34 INITIAL NL1 and NM1 (in NAMELIST GEOM) must not differ from a previous run when restarting with ISTATE>0. Remove NL1 and NM1 from NAMELIST GEOM and rerun.
- 35 OUTPUT The value I1 is an invalid value of ISTPR or NTHPR; specifically, the VV field is not defined. The value is ignored and processing continues.
- 36 INPSTR Too many partitions in Thermal Structure Prototype. Sum of NP over all NAMELISTs M for a single prototype must not exceed 49 (due to limits in Q/TSRTUC).
- 37 INPSTR The material type (MI,p.55) last printed is outside the expected range: $0 < MI \leq NMATER \leq 20$. Internally reset to $\max(1, NMATER)$ and execution continues.
- 38 INPSTR Partition size (DR) last printed must be positive. (abs(DR) has been used; exec. cont.) ?NO.MSG?
- 39 HLIQ The water property routine HLIQ was called with a temperature greater than 340.0 degrees C. This is out of the range of the current version of HLIQ. Execution terminates.
- 40 HLIQ The water property routine HLIQ has failed to converge in one hundred iterations. Execution terminates. This error is probably caused by bad values being passed as arguments to HLIQ.
- 41 TLIQ The property function TLIQ (Na,D20) has failed to converge in I1 iterations. R2 thru R5 are the given enthalpy, pressure, initial temperature guess, specific heat, and last iterate of temperature, respectively. Last iterate is returned to calling routine; exec.cont.
- 42 TSAT2 -- Reserved for use in COMMIX-2 --
- 43 TSCAN One of the following input rules for thermal structures has been violated:
NAMELISTs T can only appear first, after NAMELIST F, and after NAMELISTs M. The geometrical characteristics, IXYZ, must be one of the following values: 1, 2, 3, 11, 12, 13, 101, 102, 103. NAMELISTs F can only appear after NAMELISTs T and M. Each thermal structure must have at least one NAMELIST M. The THERMAL STRUCTURE PROTOTYPE CARDS must terminate with an 'END' card.
- 44 INPSTR When computing areas and volumes of the partitions of the thermal structure material regions, an inner radius was found to be less than $-1.0E+4 \cdot OTR$, where OTR was the outer radius specified in NAMELIST T. Check the thermal structure input for OTR, DR, and NP. If no errors are found here check the entire thermal structure prototype input. The negative radius is reset to zero and execution continues. See error #45 for a list of the variables printed above the error block.

- 45 INPSTR When computing areas and volumes of the partitions of the thermal structure material regions an inner radius was found to be larger than the outer radius. Check the thermal structure input. The inner radius is reset to the outer radius and execution continues. The parameters printed above the error block are: N - structure number, IREG - region number, IPAR - partition number, OUTR(N) - outer radius, DR - region size, ROUT - outside radius, RIN - inner radius.
- 46 TSCAN An invalid THERMAL STRUCTURE LOCATION card has been found. Either an index is out of range, or the LOC value is invalid (must be 'OUT ', 'IN ', 'END '), or the NUM value does not match the number of any THERMAL STRUCTURE PROTOTYPE.
- 47 TSCAN A THERMAL STRUCTURE PROTOTYPE has been encountered which has fluid cells interacting at both outside and inside surfaces, however, the number of cells interacting with the outside surface does not equal the number of cells interacting with the inside surface. I1, I2, I3, and I4 are the structure number, surface interaction code, number of cells interacting with surface 1, and number of cells interacting with surface 2.
- 48 TSCAN THERMAL STRUCTURE PROTOTYPE CARDS are inconsistent with the THERMAL STRUCTURE LOCATION CARDS. Either the TSP cards specify only cells interacting with the outside surface and the TSL cards specify some cells interacting with the inside surface, or the TSP cards specify only cells interacting with the inside surface and the TSL cards specify some cells interacting with the outside surface. The values printed above the error message are the same as those identified in error 47.
- 49 INPSTR The THERMAL STRUCTURE LOCATION CARD printed above the error message box indicates a cell within the ranges of the indices which is not a valid calculational cell. The specific I, J, and K indices are printed out as I9, I10, and I11. This cell is ignored and execution continues. The results which follow are likely incorrect.
- 50 INITAL New thermal structures, rebalancing regions and force structures can be input only at the beginning of a run (ISTATE=0) or the beginning of a transient (ISTATE=2). NEWTS, NEWREB, and NEWFOR have been reset to zero and execution continues.
- 51 FILLM The BOUNDARY SURFACE IDENTIFICATION CARDS have defined a single sided interior boundary surface between cells I1 and I2. Check to see that all surfaces you have defined bound calculational cells. Also be sure that any interior surface has calculational cells on both sides of it. Reread the BOUNDARY SURFACE IDENTIFICATION CARD input section and check your input. Execution continues however subsequent results are questionable.

- 52 INTURB When using the one-equation turbulence model, ALL internal velocities must be initialized to nonzero values. (Use INT.CELL.INIT.CARDS to set UL,VL,WL.)
--- NOT CURRENTLY IMPLEMENTED. REQUIRED????? ---
- 53 TSCAN Currently only 99 Thermal Structure prototypes are allowed. If more are needed, alter the value of PARAMETER PNSTRU in source and recompile code.
Execution terminates.
- 54 TSCAN Errors have been found in the order of the thermal structure prototype input. (A final catch-all!) These must be resolved before execution can continue.
- 55 INITIAL The momentum calculation scheme and the energy calculation scheme must both be run in the same mode. That is, either both must be specified implicit or both must be specified explicit; see ISYMCH and IFITEN in NAMELIST GEOM. (Use ISETMO and ISETEN in NAMELIST DATA to control frequency with which momentum and energy calculations are performed.) Also, implicit calculational schemes are required if ITURKE>1, IFPUMP>0, IFRAD>0, or IDTIME=2.
- 56 INITIAL Incorrect values have been specified for ISETEN and/or ISETMO. See NAMELIST DATA for description.
- 57 RESTAR An attempt has been made to switch from explicit to implicit mode before the completion of a timestep. Rerun an explicit restart to complete the current timestep and then run an implicit restart. ?NO.TEST?
- 58 INITIAL Using a Restart file as an initial guess (IFRES=4) is only allowed at ISTATE=0.
- 59 RESTAR The problem parameters read from the restart file for the initial guess (IFRES=4) option are inconsistent with the card input. (IMAX,JMAX,KMAX,NM1,NL1,IGEOM and IFENER were tested.)
- 60 INPUMP The PUMP LOCATION CARD printed above violates one of the following restrictions:
NAME must be 'PUMX', 'PUMY', 'PUMZ', or 'END'.
Must have $0 < N \leq NPUMP \leq 5$.
Indices (I,J,K) must be a single valid cell.
- 61 INHTX The FLUID CELL LOCATION CARD printed above violates one of the following restrictions:
NAME must be 'HTEX' or 'END'.
 $2 \leq NUM \leq IFHTX$. (1 is default; may not be input).
The range defined by (IB to IE,JB to JE,KB to KE) must refer to valid cells.
- 62 AMAIN The conjugate gradient method (ISYMCH=5) cannot be utilized with R-theta geometry (IGEOM=-1) when only 1 or 2 theta nodes are used to model a 360 degree problem. Tested are the above two flags plus JMAX (.LE.2) and the sum of the theta nodes, i.e. the sum of DY(J).GE.TESTJ. TESTJ is set internally to 355 degrees on the assumption that any problem within 5 degrees of 360 degrees was intended to be a complete cylinder. This also avoids round off problems associated with the test.
- 63 INPSTR KRAD was specified as non-zero on other than the first or last NAMELIST M for a thermal structure prototype; the input value is ignored.

64 HEATCF The heat transfer correlation index, IHTWAL or IHT, is outside of the valid range 1-20 and 51-99.

65 HEATCU The user must supply a function HEATCU to use a heat transfer correlation index, IHTWAL or IHT, of 51-99.

66 INITA2 The input associated with KFLOW=-6 has violated one of the following restrictions:
 Only 10 such surfaces allowed. (Others use KFLOW=-5)
 Must use KTEMP=400. (Set to 400)
 Should not be an IREGular SURFACE. (Warn)
 SURFACE may have only 1 SURFACE ELEMENT. (Uses the first)
 Side of adjacent cell opposite SURFACE must border a cell(warn) and other 4 sides have zero normal velocity (warn - code can not check this completely).

67 INITAL The input associated with Automatic Time Step Size Control (IDTIME=2) violates one of the following:
 Must have DTFACT>1. (Set DTFACT=2 and continue)
 Must have $0 < DT(3) \leq DT(1 \text{ and } 2) \leq DT(4)$. (Terminate)

68 TIMSTP While using the Automatic Time Step Size Control (IDTIME=2), the step size needed to be reduced below DT(3). Terminate execution; no Restart file written.

69 ERRTSC When using implicit Thermal-Structure/Fluid coupling (IHTWAL(99)=2), adjacent cells may only be coupled by a single thermal structure. Rerun the job with
 (a) new thermal structure prototypes and locations
 or (b) IHTWAL(99)<2.

70 INITAL If there are no Thermal Structures (ISTRUC=0), then you may not input IHTWAL(99)>0. Rerun job with IHTWAL(99)=0 (reorder Boundary Surfaces if needed).

71 AAMAIN Currently only 5 Fluid Types are allowed. If more are needed, alter the value of PARAMETER PIFHTX in source and recompile code. Execution terminates.

72 CHKHTX Cells containing different fluid types must be in physically separate regions. Adjust BOUNDARY SURFACES or surface permeabilities of cells and rerun.

73 INITZ1 The initialization will not work if multiple fluids are present; rerun with IFROD=1.

74 QSRAD The coefficient matrix in the radiation heat transfer matrix is singular. Check EMITS, KRAD, and FVIEW input for consistency.

75 AAMAIN Error OPENing restart or plot file.

76 RSURFO Reserved for ...
 ISURFO ... future implementation.

 * STORAGE ALLOCATION *

In order to ease the task of creating load modules (binary files) to fit the size of the problem being considered, a quasi-dynamic storage allocation scheme has been implemented. Space for most of the geometry dependent variables is allocated in the variable S of COMMON /SPACE/. The address of each variable is computed at the beginning of each run. These addresses are then passed into called subroutines where the variables are named and variably dimensioned. The total length necessary to run the problem is compared with the storage available in COMMON /SPACE/. If the available storage is inadequate, execution terminates with a message indicating the space required. By changing the dimension of S in SUBROUTINE ALTER to the value indicated, and then recompiling and relinking ALTER to the existing load module, a new load module of the required size can be obtained.

The following table shows a list of the variables with space allocated in variable S of COMMON /SPACE/. The dimension of each variable is indicated as is the index of variable IS which contains the S offset address (in words) of the variables. If any changes are made in this area, one must assure that consistency is maintained in the following areas:

ROUTINE	Comments
-----	-----
AAMAIN	Set addresses; determine space available; pass variable addresses.
TIMSTP	Subroutine call statements with arguments of the type S(IS(N)) which pass working space.
RESTAR	Statement: NEEDED=IS(250)
other	Use of variables of form S(IS(index)).

VARIABLE DIMENSION	VARIABLE NAME	INDEX	VARIABLE NAME	INDEX	VARIABLE NAME	INDEX
NM1	MIP	1	MIM	2	MJP	3
	MJM	4	MKP	5	MKM	6
	IJK	7	IDODL	# 8	IFORCE	9
	MSWEEP	# 10			TURKO	12
	ALX	13	ALY	14	ALZ	15
	AL	16	RL	17	UL	18
	VL	19	WL	20	DL	21
	HL	22	TL	23	DRDT	24
	ARUL	25	ARVL	26	ARWL	27
	REALB	28	QSOUR	29	DDDPOT	30
	DDDH	31	OARU	32	OARV	33
	OARW	34	PSTAT0	35	TURK	36
	TURVIS	37	UWIRE	38	VWIRE	39
	WWIRE	40	P	41	AC1	67
	AC2	68	AC3	69	AC4	70
	AC5	71	AC6	72	AC0	73
	BC0	74	TURCON	76	INDHTX	78

(Arrays dimensioned NM1 continue on next page.)

VARIABLE DIMENSION	VARIABLE NAME	INDEX	VARIABLE NAME	INDEX	VARIABLE NAME	INDEX
NM1-cont.	EKL	79	EKLOLD	80	TLOLD	81
	POLD	82	RLOLD	83	DRDTOL	84
	ICELL	+ 85	JCELL	+ 86	R	+ 90
	P	+ 91	F	+ 92	FF	+ 93
	BP0	+ 94	VOL	+ 95	ALNG	167
	TKED	168	TKEDO	169	BKOLD	171
	BDOLD	172	TURVIO	179		
NM1+1	MIMN	v 99	MIPN	v100	MJMN	v101
	MJPN	v102	MKMN	v103	MKPN	v104
	IREN	v105	TEMP01	v106	TEMP02	v107
	TEMP03	v108	TEMP04	v109	MOTOMR	w110
	MTOMR	w111	MTOMD	w112		
	MIMTS	144	MIPTS	145	MJMTS	146
	MJPTS	147	MKMTS	148	MKPTS	149
	NIMTS	150	NIPTS	151	NJMTS	152
	NJPTS	153	NKMTS	154	NKPTS	155
	MIME	156	MIPE	157	MJME	158
	MJPE	159	MKME	160	MKPE	161
	MIMT	v173	MIPT	v174	MJMT	v175
	MJPT	v176	MKMT	v177	MKPT	v178
NM1*6	NMNP	+ 87	AL	+ 96		
NM1*7	NA	+ 88	AU	+ 97	A	+ 98
NM1*9	NM	+ 89				
(NM1+1)*NTSCUP (NTSCUP = max # of nonadjacent cells coupled thru a thermal structure to a cell; currently 2 in version 900801 and 12 in version 910515.)						
	MTSODD	162	NTSODD	163	SPHODL	164
(NM1+1)*(NTSCUP+6)						
	QNOGHT	165	QPRIME	166		
MTP1 (= IMAX*JMAX*KMAX+1)						
	II	w113	JJ	w114	KK	w115
	MDTOMR	w116	MIMD	w117	MJMD	w118
	MKMD	w119	MIPD	w120	MJPD	w121
	MKPD	w122	MRTOMD	w123	IICEL	w124
	NDG	w125	FFF	w132	XXX	w133
	DDINV	w134	C1	w135	C2	w136
	C3	w137	C1U	w138	C2U	w139
	C3U	w140				
MTP1*6	AAL	w142				
MTP1*7	NAA	w131	AA	w141	AAU	w143
MTP1*9	NMM	w130				

VARIABLE DIMENSION	VARIABLE NAME	INDEX	VARIABLE NAME	INDEX	VARIABLE NAME	INDEX
NPLANS	(= IMAX+JMAX+KMAX-2 ... max # of diagonal planes)					
	NMAX	w127	NL	w128	NU	w129
MAXPTP*NPLANS	(where MAXPTP is max # points in a diagonal plane)					
	MINDEX	w126				
IFREB	MREB	11				
NL1	MB	42	ISURF	43	AREA	44
	RLB	45	VELBN	46	QBN	47
	HLB	48	TLB	49	PB	50
	TURKB	51	TKEDB	170		
NFORCE	ICORR	60	CLENTH	61	REYLEN	62
	FORCEF	63				
MIJK	DUMMY1	75				
NSTRUC*11	ITSCB	52				
NSTRUC	HYDRA1	53	HYDRA2	54	RODFR	55
NPAR	STAREA	56	STVOL	57		
NREG	MATERL	58	HEAT	59	SIZE	64
NSUR	ICEL	65	HSTREL	66		
NTTS	TTS	77				

In the versions of the code prepared for short-word machines (IBM,VAX), all REAL variables are declared DOUBLE PRECISION. A # in the above list indicates that storage is based on the variable being type REAL rather than INTEGER. A + in the above list indicates that the variable is used exclusively in the conjugate-gradient solution subroutines. A v in the above list indicates that the variable is used exclusively in the vectorized version of the code. A w in the above list indicates that the variable is used in the vectorized conjugate gradient momentum solution.

The INDEX above does not indicate the actual storage order within COMMON /SPACE/. This may be seen by setting ISTBUG=1 in NAMELIST GEOM. By INDEX, the variables are stored in the following order:

75; 1-8,10,13-35,41-50,78-84
 9,60-63 (only if NFORCE>0)
 11 (only if IFREB>0)
 37,76 (only if ITURKE>0)
 12,36,51,167,179 (only if ITURKE>1)
 168-172 (only if ITURKE>10)
 38-40 (only if IWIRE>1)
 52-59,64-66,77 (only if ISTRUC>0)
 67-74, 85-98 (only if ISYMCH>2 and IFITEN>2)
 99-143 (only if ISYMCH>2, IFITEN>2, and vector machine)
 144-155 (only if ISTRUC>0)
 156-161
 162-166 (only if ISTRUC>0)
 173-178 (only if ITURKE>10)
 250 (Holds the number of words actually used in /SPACE/)
 (180-249 are not currently used)

The full explanation of this storage allocation is given in Secs. 4.6 and 4.7 of Ref. 12.

 * CALLING SEQUENCE *

The following table indicates the structure of COMMIX-1AR/P by showing the calling sequence of the subroutines. Calls to the properties routines and ERRCHK are not indicated. Also multiple calls may not be indicated. An asterisk following a name indicates that the calls from that routine have been previously listed. An entry such as FACINC/FACNCV means that FACINC used on scalar machine, FACNCV used on vector machine.

```

AAMAIN
  LOCF
  CLEAR
    LOCF
  TSCAN
  -----/MXPLNS
  ALTER
    LOCF
  AMAIN
    GEOM3D
      BOXES
        FILLM
          SHOME
          TLEFTS
        QTRPIN
          FILLM
          RARRAY
          INTWIR
            WIRE
            WIRVOL
            WIRE
            RARRAY
            GETWIR
            THETAS
          FULPIN
            FILLM
            RARRAY
          TLEFTS
          MARRAY
          IARRAY
          RARRAY
        INITIAL
          NPROPS
          RESTAR
            LOCF
            PLTAPE
          INITA3
          INITA2
            LBLE
          FITIT
            SMOOTH
          GETF
          QGENER
            AXHEF
            RARRAY

```


IREBAL		
INHTX		
INPUMP		
INFORC		
INPSTR		
RESTAR		
ICTEMP		
BARIN		
	DSET3	
	RSET3	
	ISET3	
	DSET2	
	RSET2	
	ISET2	
	REDEF	
CHKHTX		
INTURK		
BCFLOT		
	GETF	
BCTEMT		
	GETF	
BCTEM0		
	GETF	
	HEATCF	
		HEATCU
	QDUCTW	
		GETF
		HEATCF *
BCTEMP		
	DUCTWA	
		GETF
		HEATCF *
BCFLOW		
BCPRES		
	GETF	
GDDDP		
GETMTS		
	GETIJK	
	ERRTSC	
	LODODD	
HSTRUC		
	HEATCF	*
TSTRUC		
	GETF	
QSTRUC		
	GETF	
	QSTRDS	
	GETF	
INITZ1		
	BCPRES	*
	BCTEM0	*
	BCTEMP	*
	BCFLOT	*
	HSTRUC	*
	TSTRUC	*
	QSTRUC	*

INITUR	TLFIX			
	INTURK			
	VISFIX			
		VELOG		
			VELCEN	
	TURV12			
OUTPUT	GETF			
	RARRAY			
	RSURFO			
	ISURFO			
	LBL			
	PSTRUC			
GDCONV				
GDDH				
PLTAPE				
TLEFTS				
WATSTP				
GDCONV				
GETEKL				
TIMSTP				
	BCPRES	*		
	MOLOOP			
		BCFLOT	*	
		XMOMI		
			WLFNCV	
				YOFRE
			FORCES	
			PUMPQ	
				GETF
		YMOMI		
			WLFNCV	*
			FORCES	
			PUMPQ	*
		ZMOMI		
			WLFNCV	*
			FORCES	
			PUMPQ	*
		PEQN		
		GETDL		
		REBAZG		
			GETDL	
			TDMA	
		REBAZ		
			GETDL	
			TDMA	
		GETDL		
		SOLVIT		
			REBAZG	*
			REBAZ	*
			GETDL	
		CGLOOP		
			STORE	
			UPDATE	
			COMMAT	
			BOUND	

```

ELIM
VOLCEL
WRPTST
SORTC
-----/MXPLNS
-----/RESORT
CNGRIC
FACINC/FACNCV
-----/COFSRT
OPERXC/OPERXD
LOWFAC/LOWFCV
UPPFAC/-----
DDOTC
DAXPYC
DAXPXC

GETDL
MOMENI
BCFLOW
VISFIX *
BCTURB
TKLOOP

TKSORC
VELCEN
TKENER
WLFNCK
YOFRE

SOLVEN/SOLVEV
TDLOOP
TDNER
SOLVEN/SOLVEV

TURV12
ENLOOP

BCTEMT *
HSTRUC *
ESORCE
GETF
BCTEM0 *
QSTRUC *

ENERGI
GETIJK
SOLVEN/SOLVEV
ENRBAL

TLEFTS
GETEKL
BCPRES *
BCTEMP *
TSTRUC *
QSRAD

DGEFA
DGESL
MALOOP

DLCALC
XMOM
FORCES
YMOM
FORCES

```

	ZMOM		
		FORCES	
	GDDDP		
	MOMENT		
	BCFLOT	*	
	BCFLOW		
	REBALX		
		MOMENT	
		BCFLOW	
	REBALY		
		MOMENT	
		BCFLOW	
	REBALZ		
		MOMENT	
		BCFLOW	
	REBAL		
		DLCALC	
		MOMENT	
		BCFLOW	
		TLEFTS	
		BCPRES	*
BCTEM0	*		
DLCALC			
GDDDH			
ENERGY			
	BCTEMT	*	
	HSTRUC		
	GETF		
	BCTEM0	*	
	QSTRUC	*	
	BCTEMP	*	
	TSTRUC	*	
BCFLOF			
	LBLF		
TLEFTS			
WATSTP			
PLTAPE			
BOIL			
WATSTP			
WATTIM			
OUTPUT	*		
TLEFTS			
RESTAR	*		
WATSTP			
OUTPUT	*		

 * OVERLAY STRUCTURE *

The versions of the code for Cray, VAX, and UNIX-based computers do not utilize an OVERLAY structure. What follows below is the OVERLAY structure used in earlier versions on IBM mainframe computers (e.g. 360/75, 370/195); as such, the structure has NOT been updated to include all routines in the current code version.

Core storage can be minimized on the IBM computers if COMMIX-1AR/P load modules are created in OVERLAY format using the following directives to the Linkage Editor:

```

INSERT AAMAIN,ERRCHK,LOCF
INSERT I4VAR,R4VAR,I4ARY,R4ARY,SPACE
INSERT MACHIN,REBALS,CCORR,CHEATC,ADDCON
INSERT ARINC,ARINI,TIMING,CPUMP,INDLIM,CSCAN
OVERLAY OVLY001
    INSERT CLEAR,TSCAN,ALTER
OVERLAY OVLY001
    INSERT AMAIN,TLEFTS,RARRAY,RESTAR,PLTAPE,BOIL
    INSERT BCFLOF,BCFLOT,BCFLOW,BCPRES,BCTEMP,BCTEMT,BCTEMO
    INSERT GETF,DUCTWA,QDUCTW,GDCONV,GDDDH,GDDDP,WATTIM,WATSTP
    INSERT CPLIQ,DRODHL,HLIQ,PSAT1,ROLIQ,THCLIQ,TLIQ,VISLIQ
    INSERT HSTRUC,QSTRUC,TSTRUC,QSTRDS,LBLE,HEATCF,HEATCU
OVERLAY OVLY003
    INSERT GEOM3D,FILLM,MARRAY,IARRAY,SHOME
OVERLAY OVLY004
    INSERT BOXES
OVERLAY OVLY004
    INSERT QTRPIN,INTWIR,WIRE,WIRVOL,GETWIR,THETAS
OVERLAY OVLY004
    INSERT FULPIN
OVERLAY OVLY003
    INSERT INITAL,INITA2,INITA3,NPROPS,FITIT,SMOOTH
    INSERT ICTEMP,BARIN,RSET3,RSET2,REDEF,INH TX,CHKHTX
OVERLAY OVLY008
    INSERT QGENER,AXHEF
OVERLAY OVLY008
    INSERT IREBAL
OVERLAY OVLY008
    INSERT INFORC
OVERLAY OVLY008
    INSERT INPUMP
OVERLAY OVLY008
    INSERT INPSTR
OVERLAY OVLY003
    INSERT INITZ1
OVERLAY OVLY003
    INSERT TIMSTP,FORCES,GETEKL,SOLVEN,SOLVEV,QSRAD

```

(Linkage Editor directives continue on the next page.)

(Continuation of Linkage Editor directives.)

```

OVERLAY OVLY014
    INSERT MOLOOP, XMOMI, YMOMI, ZMOMI, PEQN, GETDL, TDMA, WLFNCV
    INSERT REBAZG, REBAZ, SOLVIT, MOMENI, PUMPQ
    INSERT ENLOOP, ESORCE, ENERGI, ENRBAL
    INSERT CGLOOP, CNGRIC, COMMAT, FACINC, BOUND, ELIM, SORTC
    INSERT LOWFAC, UPPFAC, OPERXC, DAXPXC, DAXPYC, DDOTC
    INSERT STORE, UPDATE, VOLCEL, WRPTST
OVERLAY OVLY014
    INSERT TKLOOP, TSHEAR, TKSORC, TKENER, WLFNCK, BCTURB, TURVI1
OVERLAY OVLY014
    INSERT MALOOP, DLCALC, XMOM, YMOM, ZMOM, MOMENT, ENERGY
OVERLAY OVLY016
    INSERT REBALX
OVERLAY OVLY016
    INSERT REBALY
OVERLAY OVLY016
    INSERT REBALZ
OVERLAY OVLY016
    INSERT REBAL
OVERLAY OVLY014
    INSERT OUTPUT, RSURFO, ISURFO, PSTRUC, PSTRU1
OVERLAY OVLY014
    INSERT INTURB
ENTRY AAMAIN
NAME name

```

(End of Linkage Editor directives.)

The above OVERLAY structure assumes that the One-Equation Turbulence Model is not being used (and has not been updated to include routines associated with the Two-Equation Turbulence Model); a different structure will be more CPU-time-efficient, but require more REGION, when detailed turbulence modeling (ITURKE>1) is going to be used. The preceding section on CALLING SEQUENCE will be useful in designing other OVERLAY structures. (User-defined overlays may not be desirable under virtual-memory operating systems.)

```
*****
* MACHINE DEPENDENT ROUTINES *
*****
```

```
+-----+
| LOCF |
+-----+
```

This ANL IBM function returns the absolute core address of the variable which is passed as the argument. It is used extensively in the code to dynamically allocate storage in COMMON BLOCK /SPACE/, perform initialization, and determine the lengths of blocks to be written to the Restart File. The units of LOCF(var) after dividing by NOBPI must be "computer words". The source code for LOCF, in ASSEMBLER, will be included with IBM export versions of COMMIX. A replacement function is included in the source code for most other machines which calls system function LOC.

```
+-----+
| TLEFT |
+-----+
```

This ANL function returns the CPU time left in the current run in units of 0.01 seconds; the result thus counts downward from the time limit specified on the JOB card or QSUB card. It is used primarily in conjunction with variable TREST to determine when to write a restart file and terminate. This function is only called from function TLEFTS, which then returns the time remaining in units of seconds. The source code for TLEFT, in C, will be included with export versions for the Cray or UNIX-based systems. On other systems, and any time the code is used in an interactive environment, the functionality of this routine must be overridden by coding a function which will return a result which is greater than (a) TREST and (b) 10.0 s, after accounting for the factor of 100 difference in units.

```
+-----+
| LINPACK |
+-----+
```

The following linear algebra routines from LINPACK (J. J. Dongarra et al., "LINPACK Users's Guide", Soc. for Ind. and Appl. Math., Philadelphia (1979)) are called directly from subroutine QSRAD: DGEFA and DGED1 on short-word machines; SGEFA and SGEDI on long-word machines. LINPACK is part of /lib/libsci.a on the Cray and is available on many other machines.

Additional information on machine dependencies is given in App. C of Ref. 7 and throughout Ref. 12.

```
*****
* FINDING HOLES IN THE BOUNDARY *
*****
```

The BOUNDARY SURFACE SUMMARY is intended to aid the user in finding holes in the boundary surfaces. It is obtained by setting IBSBUG>0 in NAMELIST GEOM.

The boundary surface summary consists of two parts. First is a table of binary strings and their corresponding printed character. Following this table are JMAX planes with each calculational cell being represented by one of the characters from the first table. The binary string associated with each character indicates the location of the surface elements in the following way. Each bit in the binary string corresponds to a face of the calculational cell. The first bit corresponds to the face in the I minus (I-) direction. This is the surface between cell (I,J,K) and cell (I-1,J,K). The second bit corresponds to the face in the I+ direction, the third in the J-, the fourth in the J+, the fifth in the K-, and the sixth in the K+ direction. A surface element is defined at a cell face if the bit corresponding to that face has a value of 1.

For example, suppose "F" is printed at the location for cell location (I,J,K). "F" corresponds to the binary string "011000". This indicates that a surface element has been defined in the I+ and J- directions, that is, between cells (I,J,K) and (I+1,J,K) and between cells (I,J,K) and (I,J-1,K).

In order for this scheme to be effective the table should contain 62 different printable characters: (1) a "blank" corresponds to string "000000" and (2) string "111111" should never occur. While we have a laser printer with both upper and lower case at ANL, the printers usually used are impact printers with only about 58 different characters; therefore, the current implementation uses the character "?" to correspond to all of the following binary strings: "111110", "111101", "111011", "110111", "101111", "011111", and "111111". This does introduce some ambiguity; however, the impact is probably not serious. For those users who wish to eliminate these duplications, changes must be made in SUBROUTINE SHOME.

Note that this summary only shows BOUNDARY SURFACES as defined by input in NAMELIST GEOM and BOUNDARY SURFACE IDENTIFICATION CARDS; it does not show zero-permeability cell faces as set by inputting ALX, ALY, and ALZ on INTERNAL CELL INITIALIZATION CARDS.

 * Appendix B *
 * INPUT VARIABLE INDEX *

As an aid to finding input variables in the above input description, an index is provided here. A small description is given only as a variable reminder; it is not intended to replace the actual input description. ('Orphans' are not shown.)

VARIABLE	PAGE NO.	NAMELIST or CARD TYPE	DESCRIPTION
ACORP(5)	48	PUMP	Pres Incr Eq Coef (IFPUMP=1)
ACORRL(20)	39	DATA	Fric Fact Correl Coef for Re<REYTRN
ACORRT(20)	39	DATA	Fric Fact Correl Coef for Re>REYTRN
AKAPPA	43	DATA	Turb. const. (Von Karmon)
AL	59	ICIC	Cell Volume Porosities(fluid fraction)
ALPHA	43a	DATA	Full/Semi Implicit time adv (1/0)
ALPHAL(5)	48	PUMP	Exp. for lam. flow (IFPUMP=1)
ALPHAT(5)	48	PUMP	Exp. for turb. flow (IFPUMP=1)
ALX	59	ICIC	Cell I-dir. Surface Porosity
ALY	59	ICIC	Cell J-dir. Surface Porosity
ALZ	59	ICIC	Cell K-dir. Surface Porosity
AO	43a	DATA	Donor/Central flux (explicit)
AREA	14, 58	BSIC, BVIC	Surface Element Areas
BCORP(5)	48	PUMP	Pres Incr Eq Coef (IFPUMP=1)
BCORRL(20)	39	DATA	Fric Fact Correl Coef for Re<REYTRN
BCORRT(20)	39	DATA	Fric Fact Correl Coef for Re>REYTRN
BO	43a	DATA	Donor/Central flux (explicit)
C0CP(20)	28	DATA	Specific Heat Coeffs. for solids
C1CP(20)	28	DATA	Specific Heat Coeffs. for solids
C2CP(20)	28	DATA	Specific Heat Coeffs. for solids
C0K(20)	28	DATA	Conductivity Coeffs. for solids
C1K(20)	28	DATA	Conductivity Coeffs. for solids
C2K(20)	28	DATA	Conductivity Coeffs. for solids
C0RO(20)	28	DATA	Density Coeffs. for solids
C1RO(20)	28	DATA	Density Coeffs. for solids
C2RO(20)	28	DATA	Density Coeffs. for solids
CCORP(5)	48	PUMP	Pres Incr Eq Coef (IFPUMP=1)
CCORRL(20)	39	DATA	Fric Fact Correl Coef for Re<REYTRN
CCORRT(20)	39	DATA	Fric Fact Correl Coef for Re>REYTRN
CDRAGX	42	DATA	Fuel assy drag force multiplier
CDRAGY	43	DATA	Fuel assy drag force multiplier
CDRAGZ	43	DATA	Fuel assy drag force multiplier
CDTURB	42b	DATA	Coef. in turb. viscosity calc.
CHARRE	42	DATA	Charact. Reynolds # in Turb.
CHART	42	DATA	Charact. T in Turbulence Model
CHGMAX	21	DATA	Up limit change test (IDTIME=2)
CHGMIN	21	DATA	Low limit change test (IDTIME=2)
CLADOD	13, 45	GEOM, INPUTQ	Fuel pin diameter
CLENTH(20)	39	DATA	Charact. Lengths, Force Struct.

Variable	Page	Type	Description
CORPF(9)	49	PUMP	Frict. Torque Coef.(IFPUMP=3&4)
CORPH8(9)	49	PUMP	Head Coef.(IFPUMP=3&4)
CORPT(8)	49	PUMP	Hyd. Torque Coef.(IFPUMP=3&4)
CORPT8(18)	49	PUMP	Hyd. Torque Coef.(IFPUMP=3&4)
CT1	43	DATA	Empirical const. in TKED eq.
CT2	43	DATA	Empirical const. in TKED eq.
CUTPF(2)	49	PUMP	Speed bounds: frict.torq.(IFPUMP=3&4)
CWIREI	13	GEOM	Interior wire wrap scale factor
CWIREO	13	GEOM	Wall wire wrap scale factor
CWIREX	43	DATA	X-Coeffs. of wire force
CWIREY	43	DATA	Y-Coeffs. of wire force
CWIREZ	43	DATA	Z-Coeffs. of wire force
DDHMX	43a	DATA	Explicit Solution DDDH
DENRO	50	PUMP	Density*Gravity for Fluid Being Pumped
DLCUT	43a	DATA	Eliminate Cells from Mass/Mom. Iter.
DR	55	TSP-M	Partition size in thermal structure
DT(4)	20,21	DATA	Time Step Sizes
DTFACT	21	DATA	Time Step Change Factor (IDTIME=2)
DX(99)	12	GEOM	I-direction cell sizes
DY(99)	12	GEOM	J-direction cell sizes
DZ(99)	12,13	GEOM	K-direction cell sizes
EMITS(99)	40	DATA	Emissivity of Radiating Surface
EPS1	22	DATA	Conv. criterion: mass residual
EPS2	22	DATA	Conv. criterion: mass residual
EPS3	22	DATA	Conv. criterion: UL,VL,WL,HL
EPS5	22	DATA	Conv. criterion: energy inners
EPS6	42b	DATA	Conv. criterion: turb. inners
FC0H(5)	31	DATA	Fluid Enthalpy Coef.
FC1H(5)	31	DATA	Fluid Enthalpy Coef.
FC0K(5)	31	DATA	Fluid Conductivity Coef.
FC1K(5)	31	DATA	Fluid Conductivity Coef.
FC0MU(5)	31	DATA	Fluid Viscosity Coef.
FC1MU(5)	31	DATA	Fluid Viscosity Coef.
FC0RO(5)	31	DATA	Fluid Density Coef.
FC1RO(5)	31	DATA	Fluid Density Coef.
FC0T(5)	calculated by code		Fluid Temp Coef.(H-inverse)
FC1T(5)	calculated by code		Fluid Temp Coef.(H-inverse)
FCTHI	31	DATA	Max. Temp. for fluid prop. edit
FCTLO	31	DATA	Min. Temp. for fluid prop. edit
FNZ	44	INPUTQ	Axial Hot Channel Factor for Axial Q
FORCEF(20)	39	DATA	Force Structure Coefficients
FVAL(100)	33	DATA	Transient driving function points
FVIEW(99,99)	40	DATA	Radiation View Factor from L to M

Variable	Page	Type	Description
GASMW(5)	31	DATA	Fluid Molecular Weight (Ideal Gas)
GRAVX	25	DATA	Gravity vector component, X-dir
GRAVY	25	DATA	Gravity vector component, Y-dir
GRAVZ	25	DATA	Gravity vector component, Z-dir
HEATC1(20)	27	DATA	Nusselt No. correlation Coef.
HEATC2(20)	27	DATA	Nusselt No. correlation Coef.
HEATC3(20)	27	DATA	Nusselt No. correlation Coef.
HEATC4(20)	27	DATA	Nusselt No. correlation Coef.
HGAP	55	TSP-M	Gap Material Heat Transfer Coef.
HL	59	ICIC	Enthalpy by Cell
HLB	58	BVIC	Enthalpy by Boundary Surf Element
HSINK(99)	26	DATA	Heat Trans. Coeff. to sink (WallModel)
HTEX	47	FCLC	Fluid No. in each Internal Cell
HYD	55	TSP-F	Hyd Dia for Fluid/Str Heat Trf.
HYDIN	42a	DATA, INPUTQ	Hydraulic Dia.: Turb & Asy. Models
HYDWAL(99)	26	DATA	Hydraulic Dia, by Surf. (WallModel)
IBOIL	17	DATA	Check for Boiling Flag(0=no check)
IBSBUG	12(19)	GEOM	Boundary Surface Summary Print Flag
ICORP(5)	48,49	PUMP	Correl # for each Pump
ICORR(20)	39	DATA	Force Correlation Types(User&Library#)
IDDDP	43a	DATA	Explicit DDDP Flag
IDLBUG	18	DATA	Difference Eqtns. Debug Flag
IDRAG	43	DATA	Fuel Assy Drag Flag (0=off)
IDTIME	20	DATA	Time Step Size Selection Flag
IENBUG	18	DATA	Energy & Radiation Debug Flag
IFENER	17	DATA	Energy Calc. Flag(0=off, 1 or 2 = on)
IFHTX	9	GEOM	Multifluid Flag: # of fluids
IFITEN	9	GEOM	Energy Calc Solution Technique Flag
IFPROP	17	DATA	Fluid #1 Prop.Flag(0=BuiltIn;1=ReadIn)
IFPUMP	41	DATA	Pump Model Flag(0=no pump)
IFRAD	40	DATA	Radiation Heat Transfer Flag(0=off)
IFREB	10	GEOM	Rebalancing Flag(= # of rebal. cell+face)
IFRES	10-11	GEOM	Restart Flag(0-4=no/W/R/RW/RGW)
IFROD	17	DATA	Fuel Rod Flag(0 = no)
IGEOM	8	GEOM	Geom Flag (0/-1/>0 = Box/Cyl/Hex)
IHT	55	TSP-F	Index for Fluid/Str Ht.Trf.Coeff.
IHTWAL(99)	26	DATA	Heat Transfer Correl, by Surf. (WallModel)
IHTWAL(99)	17	DATA	Loc. 99 only: T.S. soln. method
IINBUG	18	DATA	Hex Can Option Debug Flag
IJTYPE	45	INPUTQ	Cell Type for Pins
IMAX	12	GEOM	Max # of Cells, X(R)-direction
IMOBUG	19	DATA	Debug Flag for Momentum Solution
IN	56	TSLC	Inner Surface Location of T.S.
INDHTX	47	FCLC	Cell Fluid # identifier
IPART	13	GEOM	Pin Partition Flag(0=quarter;1=full)
IPTRAN	50	PUMP	Pump Speed Integration Model
IPUBUG	50(19)	PUMP	Pump Model Debug Flag
IQ	44	INPUTQ	Axial Power Distribution Control
IREBIT	30	DATA	Rebalance Iteration Interval
IREG	14	BSIC	Identify IREGular Surface Elements

Variable	Page	Type	Description
ISETEN	16	DATA	Energy Eqtn. Periodic Use Flag
ISETMO	16	DATA	Momentum Eqtn. Periodic Use Flag
ISTATE	16	DATA	0/1/2/3=SS/SScont/Trans/TranCont.
ISTBUG	12(19)	GEOM	Storage Layout Table Print Flag
ISTPR(50)	36-37	DATA	Print Opt.: arrays printed initially
ISTRUC	11	GEOM	Thermal Structures Presence Flag(0=none)
ISYMCH	9	GEOM	Momentum Solution Technique Choice
ITAUTO	(20)	(internal)	1=activated Auto DT under IDTIME=2
ITDEBUG	19	DATA	Turb.Model (TKED) Debug Flag
ITIBUG	18	DATA	Pressure Solution Debug Flag
ITKBUG	19	DATA	Turb.Model (TURK) Debug Flag
ITMAXE	22	DATA	Max # of Energy Iterations (inner)
ITMAXK	42b	DATA	Max # of Turb.KE Iterations (inner)
ITMAXP	22	DATA	Max # of Pressure Iterations (inner)
ITSBUG	53(19)	STRUCT	Thermal Structure Debug Flag
ITURKE	11(42)	GEOM	Turbulence Model Flag(0 = off)
IT(2)	22	DATA	# of Iterations per time step (outer)
IWIRE	13	GEOM	Wire Wrap Flag(0 = off, 2 = exact)
IXMBUG	19	DATA	Debug Flag
IXREB	29	DATA	X-Direction Rebalance Flag
IXYZ	54	TSP-T	T.S. Geometry Type
IYMBUG	19	DATA	Debug Flag
IYREB	29	DATA	Y-Direction Rebalance Flag
IZMBUG	19	DATA	Debug Flag
IZREB	29	DATA	Z-Directional Rebalancing Flag
JMAX	12	GEOM	Max # of Cells, Y(THETA)-direction
KFLOW(99)	23	DATA	Type of Surface Velocity B.C.
KHHS	44	INPUTQ	Highest Heated K-Plane
KLHS	44	INPUTQ	Lowest Heated K-Plane
KMAX	12,13	GEOM	Max # of Cells, Z-direction
KPRES(99)	24	DATA	Type of Surface Pressure B.C.
KPROP(99)	24	DATA	Fluid # for each Surface
KRAD	55	TSP-M	Radiating Surface Number
KTEMP(99)	24	DATA	Type of Surface Temperature B.C.
LASTDT	20	DATA	Time Step # to switch from DT(1) to DT(2)
LASTIT	22	DATA	Time Step # to switch from IT(1) to IT(2)
LMPRNT	11(19)	GEOM	Cell&Surface Numbering Edit Flag
LOC	56	TSLC	T.S. Location cards (=IN/OUT/END)
MATWAL(99)	26	DATA	Wall Mat. Type,by Surf. (WallModel)
MI	55	TSP-M	Material Type Index
N	54	TSP-T	Thermal Structure Prototype Number
NCORR	39	DATA	# of Correlation Types for Force Structure
NCORRP	48,49	PUMP	Total # of Pump Correls.
NCUT	50	PUMP	Pump Speed Threshold for Low Flow Correl.
NEND(25)	33	DATA	# Pts in each Transient Function
NEWFOR	16	DATA	New Force Structure input Flag
NEWREB	16	DATA	New User Rebalance input Flag
NEWTs	16	DATA	New Thermal Structure input Flag
NFORCE	11	GEOM	# of Force Structures
NHEATC	27	DATA	# of Fluid/Structure Heat Trans. correls.

Variable	Page	Type	Description
NL1	8	GEOM	Total # of Surface Elements
NM1	8	GEOM	Total # of Cells
NMATER	28	DATA	# of Solid Materials (Wall, TS Models)
NOFQT	33, 44	DATA, INPUTQ	Transient Function # for Heat Source
NP	55	TSP-M	No. of Partitions in T.S.
NPUMP	48, 49	PUMP	# of Pumps
NRAD	40	DATA	No. of Radiating Surfaces
NREBM(99)	29	DATA	# of Cells in each Rebal Region
NREBRT	29	DATA	# of User Defined Rebalance Regions
NREBX(99)	29	DATA	# of X surfaces between Rebal Regions
NREBY(99)	29	DATA	# of Y surfaces between Rebal Regions
NREBZ(99)	29	DATA	# of Z surfaces between Rebal Regions
NSURF	12	GEOM	# of unique (direction, BC, prop) surfaces
NT	54	TSP-T	Trans. Func. # for T.S. Heat Source
NTF	33	(DATA)	Trans. Func. # (not input by user)
NTHCON(11)	21	DATA	Time step #'s at which conv. is checked
NTHPR(50)	36-37	DATA	Print Opt.: selects arrays to be edited
NTJUMP	42b	DATA	Freq. at which Turb. Eq. solved
NTMAX	20	DATA	Max Time Step # Allowed
NTORQM(5)	50	PUMP	Trans. Functn. # applied to motor torque
NTOTS(99)	33	DATA	Thermal Structure Heat Source: NT override
NTPLOT(26)	34	DATA	Time Step Control of Plotting Option
NTPRNT(51)	35	DATA	Print Option Via Time Step No.
NTPUMP(5)	48, 50	PUMP	Trans. Fctn. # applied to pump speed/head
NTSCUP	56	(code)	PARAMETER: max # TS couple cells
OMEGA	22	DATA	Relaxation Factor: Pressure Solution
OMEGAD	43	DATA	Relaxation Factor: TKED eq.
OMEGAE	22	DATA	Underrelaxation Factor: Energy Eqtns.
OMEGAK	42b	DATA	Relax Factor: Turb K.E. Eq. Coeffs.
OMEGAT	42a	DATA	Relaxation Factor: TURK eq.
OMEGAV	22	DATA	Underrelaxation Fac.: Momentum Eqtns.
OMR	40	DATA	Underrelaxation Fac.: Radiation Model
OUT	56	TSLC	T.S. Outer Surface Location
OUTR	54	TSP-T	Thermal Structure Outer Radius
P	59	ICIC	Pressure (above static) by Cell
PB	58	BVIC	Boundary Pressure by Surf Element
PIFHTX	9	(code)	PARAMETER: max # Fluids
PINAF(50)	45	INPUTQ	Fraction of Pins in a Cell Type
PINER(5)	50	PUMP	Pump Moments of Inertia
PITCH	13	GEOM	Pin Pitch (nominal ctr-ctr)
PITCHX	45	INPUTQ	Pin Pitch (ctr-ctr in X-dir)
PITCHY	45	INPUTQ	Pin Pitch (ctr-ctr in Y-dir)
PNRAD	39	(code)	PARAMETER: max # Rad. Surfs.
PNSTRU	50	(code)	PARAMETER: max # Therm. Struc.
PRES(99)	25	DATA	Initial uniform Boundary Pressure
PRES0(5)	25	DATA	Fluid Reference Pressure
PRNDLD	43	DATA	Prandlt #: visc. in TKED eq.
PRNDLH	42a	DATA	Prandlt #: turb. conductivity
PRNDLK	42b	DATA	Prandlt #: visc. in TURK eq.
PUMLEN(5)	48	PUMP	Eq Dia in Re # for turb/lam trans.
PUMPHR(5)	49	PUMP	Reference Pump Heads
PUMPS(5)	48, 49	PUMP	Instantaneous Pump Speeds
PUMPSR(5)	48, 49	PUMP	Reference Pump Speeds
PUMX	51	PLC	Pumped fluid motion: I-direction
PUMY	51	PLC	Pumped fluid motion: J-direction
PUMZ	51	PLC	Pumped fluid motion: K-direction

Variable	Page	Type	Description	89
Q	55	TSP-M	Volumetric Heat Source in T.S. material	
QBN	58	BVIC	Boundary Heat Flux by Surf Element	
QCUT	50	PUMP	Low Flow Cutoff Mass Flow Rate	
QFLOWR(5)	48,49	PUMP	Ref. Pump Volume Flow Rates	
QFLUX	44	INPUTQ	Avg. Const Heat Flux	
QIJ(10,10)	26,44	(code)	--- See variable "QIN". ---	
QIN(100)	44	INPUTQ	Normalized Radial Q Distribution	
QK(99)	26,44	DATA,INPUTQ	"Axial" Heat Factor (cell average)	
QSCOOOL	44	INPUTQ	Volumetric Heat Source for Coolant	
QSOUR	59	ICIC	Volumetric Heat Source for cells	
RDTIME	21	DATA	Auto Time Step Factor(IDTIME=1 only)	
REBM	46	RRC	Rebalancing Cells	
REBX	46	RRC	I-Direction Rebalancing Surfaces	
REBY	46	RRC	J-Direction Rebalancing Surfaces	
REBZ	46	RRC	K-Direction Rebalancing Surfaces	
REG	14	BSIC	Identify REGular Surface Elements	
RELAXE	22	DATA	Relaxtion Factor: Energy Solution	
RELAXK	42b	DATA	Relaxation Factor: Turb. Soln.	
REYLEN(20)	39	DATA	Lengths for Re# for Force Structures	
REYTRN(20)	39	DATA	Transition Re # for Friction Correls	
REYTRP(5)	48	PUMP	Pump Transition Re #'S	
RLB	58	BVIC	Boundary Density by Surf Element	
RODFR	54	TSP-T	# of repetitions/cell of T.S. prototype	
SGAP	55	TSP-M	Gap Size between material regions in T.S.	
TDIN	42b	DATA	Coef. Inlet Turb. KE Dissipation	
TEMP(99)	25	DATA	Initial uniform Boundary Temp/Heat-Flux	
TEMP0(5)	25	DATA	Fluid Reference Temperature	
TIMAX	20	DATA	Max Transient Time Allowed in Simulation	
TKIN	42b	DATA	Coef. Initial/Inlet Turb. KE	
TL	59	ICIC	Temperature by Cell	
TLB	58	BVIC	Boundary Temperature by Surf Element	
TORQFR(5)	49	PUMP	Rated Frictional Torques	
TORQM(5)	50	PUMP	Rated Motor Torques	
TORQR(5)	49	PUMP	Rated Hydraulic Torques	
TPRNT(51)	35	DATA	Print Option Via Time(sec.) into Calc.	
TREST	20	DATA	CPU Sec. Left to Trigger End of Simulat.	
TSINK(99)	26	DATA	Temperatures of Sinks (WallModel)	
TSTART	20	DATA	Transient Start Time	
TURBC	42	DATA	Turbulent Conductivity	
TURBV	42	DATA	Turbulent Viscosity	
TURC	59	ICIC	Turbulent Conductivity by cell	
TURV	59	ICIC	Turbulent Viscosity by cell	
TVAL(100)	33	DATA	Transient Driving Function Time Points	
UL	59	ICIC	I-dir Velocity by Cell	
VELB	58	BVIC	Boundary Velocity by Surf Element	
VELOC(99)	25	DATA	Initial uniform Boundary Velocity by Surf	
VL	59	ICIC	J-dir Velocity by Cell	
WALLCL	13	GEOM	Clearance Between Pin Wall & Duct	
WALLDX(99)	26	DATA	Wall Thickness by Surf (WallModel)	
WALLQS(99)	26	DATA	Vol. Heat Sources by Surf (WallModel)	
WETLN(50)	45	INPUTQ	Wetted Length by Cell Type	
WIREP	13	GEOM	Wire Wrap Pitch	
WL	59	ICIC	K-dir Velocity by Cell	
WODIN	13	GEOM	Wire Wrap OD(central hex can region)	
WODOUT	13	GEOM	Wire Wrap OD(next to duct)	

Variable	Page	Type	Description
XFOR	52	FSLC	I-Direction Force Location
XNORML(99)	12	GEOM	Surface Normal: I-Direction Component
XPRES0(5)	25	DATA	X-coordinate of Fluid Pres.Ref.Pt.
YFOR	52	FSLC	J-Direction Force Location
YNORML(99)	12	GEOM	Surface Normal: J-Direction Component
YPRES0(5)	25	DATA	Y-coordinate of Fluid Pres.Ref.Pt.
ZAT0	13	GEOM	Wire Wrap Ref. Z-Height
ZFOR	52	FSLC	K-Direction Force Location
ZNORML(99)	12	GEOM	Surface Normal: K-Direction Component
ZPRES0(5)	25	DATA	Z-coordinate of Fluid Pres.Ref.Pt.

Codes for Input Block Type:

BSIC : Boundary Surface Identification Cards
 BVIC : Boundary Value Initialization Cards
 DATA : NAMELIST DATA
 FCLC : Fluid Cell Location Cards
 FSLC : Force Structure Location Cards
 GEOM : NAMELIST GEOM
 ICIC : Internal Cell Initialization Cards
 INPUTQ: NAMELIST INPUTQ
 PLC : Pump Location Cards
 PUMP : NAMELIST PUMP
 RRC : Rebalancing Region Cards
 STRUCT: NAMELIST STRUCT
 TSLC : Thermal Structure Location Cards
 TSP-F : Thermal Structure Prototype - NAMELIST F
 TSP-M : Thermal Structure Prototype - NAMELIST M
 TSP-T : Thermal Structure Prototype - NAMELIST T

```

*****
*           Appendix C           *
* GRAPHICAL OUTPUT CAPABILITIES *
*****

```

The COMMIX code does not produce graphical output directly; instead, the code writes two files of use in producing figures:

- Plot File (UNIT=76,FILE=PLTDAT): contains velocities and temperatures at time steps during the COMMIX run as specified in array NTPLOT.
- Restart File (UNIT=10,FILE=CMXNEW): contains all variables at the last time step in a single COMMIX run if IFRES=1 or IFRES=3; appended to this are velocities and temperatures in Plot File format for the last time step, regardless of the NTPLOT array.

There are four programs which post-process these files to produce graphical output:

Program	Input	Output
VECT	Plot Restart	Draw vectors superposed on planar cut of geometry to represent magnitude and direction of mass flow.
TEMP	Plot Restart	Draw numerals or isolines superposed on planar cut of geometry to represent temperature.
TDSC	Plot	A user-selected subset of all velocities and temperatures reformatted for input to GRPH.
GRPH	TDSC	Draw plots of selected variables versus time.

VECT and TEMP can read either COMMIX file as input, TDSC can only read a Plot file, and GRPH can read neither COMMIX file but can read the file written by TDSC. TDSC does not draw any figures; the other three do.

All programs are written in FORTRAN-77. The primary non-standard feature is use of NAMELIST input (as is true of COMMIX). Storage is allocated within Common Block /SPACE/ on a quasi-dynamic basis; the dimension of array S in Subroutine ALTER sets the total amount of work space available. Input Guides are an integral part of the source coding and are nominally printed during execution (unless MENUE=0 is input). The programs may be run on Cray, IBM, or Sun computers; however, the input binary file must be in the format of the machine on which the code is being run. (A fifth program is available to convert Plot and Restart files from Cray to IBM or IEEE format and vice versa.) All drawing is done via calls to routines in the DISSPLA software package, which is copyright by Computer Associates International, Inc. (711 Stewart Avenue, Garden City, NY 11530-4787, 516-227-3300). Figures are nominally sized to fit 8.5 by 11 inch paper with a 1 inch border. Graphical output from VECT, TEMP, and GRPH is in the form of a device-independent DISSPOP MetaFile, which must be further post-processed by calling a device-dependent system routine to obtain viewable output.

Sample jobstreams for using the graphics post-processors at Argonne National Lab are noted in Appendix E - USING COMMIX AT ANL. The sample jobstreams must be modified to specify proper user files and drawing directives before being submitted. Additional documentation may be found in Appendices C and D of Ref. 7.

In addition to the use of DISSPLA, several other site-dependent routines are invoked by the graphics post-processors:

IBM	Cray	Description
-----	-----	-----
LOCF	LOC	Function to return core address, in bytes on IBM and words on Cray, of the variable specified as an argument. (Called from Subroutine ALTER; see "MACHINE DEPENDENT ROUTINES" in this Guide.)
DATE1	DATE	Returns current date as "mm/dd/yy".
TIME1	CLOCK	Returns current time of day as "hh:mm:ss".
ANLACT	(sh)	ANL subroutine to return various user identification from JOB card for extra figure labelling. (Similar information from shell variables is passed through ALTER on the Cray.)

 * Appendix D *
 * REFERENCES *

1. W. T. Sha, H. M. Domanus, R. C. Schmitt, J. J. Oras, and E. I. H. Lin, "COMMIX-1: A Three-Dimensional Transient Single-Phase Component Computer Program for Thermal-Hydraulic Analysis", NUREG/CR-0785 (or ANL-77-96), Argonne National Laboratory, Argonne, IL (September 1978).
2. H. M. Domanus, W. T. Sha, R. C. Schmitt, and V. L. Shah, "COMMIX-1A: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume I: Users Manual", NUREG/CR-2896 (or ANL-82-25), Argonne National Laboratory, Argonne, IL (December 1983).
3. H. M. Domanus, R. C. Schmitt, W. T. Sha, and V. L. Shah, "COMMIX-1A: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume II: Assessment and Verification", NUREG/CR-2896 (or ANL-82-25), Argonne National Laboratory, Argonne, IL (December 1983).
4. "COMMIX-1B: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume I: Equations and Numerics", NUREG/CR-4348 (or ANL-85-42), Vol. I, Argonne National Laboratory, Argonne, IL (September 1985).
5. "COMMIX-1B: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume II: User's Manual", NUREG/CR-4348 (or ANL-85-42), Vol. II, Argonne National Laboratory, Argonne, IL (September 1985).
6. R. N. Blomquist, P. L. Garner, and E. M. Gelbard, "COMMIX-1AR/P: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume 1: Equations and Numerics", ANL-90/45, Argonne National Laboratory, Argonne, IL (July 1991).
7. P. L. Garner, R. N. Blomquist, and E. M. Gelbard, "COMMIX-1AR/P: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume 2: User's Guide", ANL-92/33, Argonne National Laboratory, Argonne, IL (September 1992).
8. J. F. Koenig, unpublished information, Argonne National Laboratory, Idaho Falls, ID (November 23, 1982).

9. Imtiaz K. Madni and Erik Cazzoli, "A Single-Phase Pump Model for Analysis of LMFBR Heat Transport Systems", NUREG/CR-0240 (or BNL-NUREG-50859), Brookhaven National Laboratory, Upton, NY (June 1978).
10. F. F. Chen, unpublished information, Argonne National Laboratory, Argonne, IL (April 27, 1983).
11. R. N. Blomquist, P. L. Garner, and E. M. Gelbard, "COMMIX-1AR/P: Program for Three-Dimensional Transient Thermal Hydraulics," National Energy Software Center, Argonne National Laboratory, Argonne, IL, NESC #9450 (August 1989). [This is the APR1789 version, which does not have k-epsilon turbulence model; not cited.]
12. P. L. Garner, R. N. Blomquist, and E. M. Gelbard, "COMMIX-1AR/P: A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems; Volume 3: Programmer's Guide", ANL-92/34, Argonne National Laboratory, Argonne, IL (September 1992).

```
*****
*      Appendix E      *
*  USING COMMIX AT ANL  *
*****
```

The COMMIX-1AR/P code and the four post-processor codes for graphical output may be executed on either the Cray X-MP/18 or IBM 3084 mainframe computers at Argonne National Laboratory. A fifth post-processor which converts Restart and Plot binary format files from Cray to IBM and vice versa may be executed only on the Cray. Use of the Cray, rather than IBM, version is generally preferred due to run time considerations. All programs are normally run in batch, rather than interactive, mode.

To run any of these codes, the user must compile subroutine ALTER to set the size on COMMON BLOCK /SPACE/, link this with the other routines from a main object library, and provide "card" and "binary" file input for execution. This process is simplified on the Cray by the use of a set of shell scripts.

A set of sample jobstreams has been prepared to aid the user. The sample to follow depends on the system to which and from which the job is submitted. In the names that follow, "zzzz" may be any of the graphics post-processors: VECT, TEMP, TDSC, or GRPH. The sample jobs must be modified before being submitted; follow instructions embedded in each sample. Useful files and scripts:

- Run on Cray submitted from IBM (Wylbur): see members in partitioned dataset B27484.COMMIX.CRSCRIPT for samples:
 - US900801 : execute COMMIX
 - PzzzzUSE : execute graphics post-processor zzzz
 - FILEUSE : execute file converter
- Run on Cray submitted from Central VAX Cluster: see files in directory CC116:[B27484.CMXCRGEN] for samples:
 - US900801.TCJOB : execute COMMIX
 - PzzzzUSE.TCJOB : execute graphics post-processor zzzz
- Run on IBM submitted from IBM (Wylbur): see members in partitioned dataset B27484.COMMIX.IBMSAMP for samples:
 - US900801 : execute COMMIX
 - PLOTzzzz : execute graphics post-processor zzzz
- The scripts used on the Cray are self-documenting if a question mark, ?, is entered as the only argument:
 - /n1/b27484/cmx900801 ? # COMMIX execution info
 - /n1/b27484/cmxPLOTzzzz ? # graphics program zzzz info
 - /n1/b27484/cmxFILE ? # file converter execution info

Version 910515 has NTSCUP=12, TURVIS change affects DTIME under IDTIME=2, and miscellaneous fixes. Access by changing "900801" to "910515" in the above file names.

Selected members of the COMMIX-1AR/P program system are being made available on the Reactor Analysis (RA) Computing Network at ANL. The status of this implementation (e.g. which programs are available, how to use them) is reviewed by executing '/home/pepper2/b27484/CMXUSER/news', which is a script file which interactively provides you with a list of current COMMIX news topics and prompts you for a topic on which you would like further information.

The master copy of this input guide is stored on the IBM MVS batch disks as sequential file 'B27484.COMMIX.MANUAL.D900801' in Wylbur EDIT format. (An up-to-date copy is also on the RA Computing Network as file '/home/pepper2/b27484/CMXUSER/ingide.910515'.) A "1" in column 1 signals the start of a new page; column 1 is blank in all other lines; maximum line length is 80 characters. Each page contains a maximum of 58 printable lines.

- To get paginated listing at destination 'dest':
 - From Wylbur:


```
USE $B27484.COMMIX.MANUAL.D900801 CLE
LIST F/L UNNUM OFFLINE DEST dest MARKER 1
```

 where you must change dest (e.g. 3800, which is the IBM model 3800 page printer in Bldg. 221 at ANL).
 - From an IBM Batch job: submit the following JCL:


```
//CMXIG JOB (alias,Fnnnnn),USER=Bnnnnn,PASSWORD=racfpassword
//*MAIN LINES=20
//*FORMAT PR,DDNAME=,COPIES=0
//*FORMAT PR,DDNAME=LIST.SDSKLIST.SYSUT2,COPIES=1,DEST=dest
//UNPR EXEC WUNPRESS,OPTIONS='UNN',
//      TSODSN='&TEMPLIS',TSODSP2=PASS,TSOUNIT=SASCR,
//      WYLDN='B27484.COMMIX.MANUAL.D900801'
//LIST EXEC SDSKLIST,INDSN='&TEMPLIS',PREFORM=A,
//      INDSP1=OLD,INDSP2=DELETE,INDSP3=DELETE,INUNIT=SASCR
```

 where you must change alias (e.g. 'null'), nnnnn (i.e. 5-digit badge number), racfpassword, and dest (e.g. 3800).
 - From RA Computing Network:


```
lpr -Pdest -CFnnnnn -f /home/pepper2/b27484/CMXUSER/ingide.910515
```

 where you must change dest (e.g. 3800) and nnnnn (i.e. 5-digit badge number). CAUTION: The '-f' option of 'lpr' is needed to get nicely paginated output; however, you can not use the '-f' option of 'lpr' for the laser printers on the RA Network (e.g. dest=1w5).
- To bring a copy to another machine: see instructions in:
 - VAX: EDIT CC116:[B27484.CMXCRGEN]MANTOVAX.IBMJOB
 - Cray: /n1/b27484/cmxMANTOCray ?

APPENDIX B: DETAILED FLUID PROPERTY EXAMPLES

As described in Sec. 9.1, fluid number 1 in a COMMIX calculation uses "detailed" fluid properties when the user sets the value of input variable IFPROP to 0 in NAMELIST DATA. Detailed properties are computed from a set of FUNCTION subprograms, which must be compiled and linked with the rest of the coding in COMMIX for execution. The properties to be provided and their units are shown in Table 9.2 along with the corresponding FUNCTION names and their allowed arguments.

Detailed property packages have been prepared, and are distributed with the COMMIX source code, for liquid sodium, liquid normal water, liquid heavy water, and helium gas; the specific functional forms used in each of these fluid packages are given in the following subsections. Only one of these fluid property packages may be linked with the rest of the coding in COMMIX during a calculational history. The nomenclature and units defined in Table 9.2 (e.g. ρ is density [kg/m³] and T is temperature [°C]) are used throughout the following subsections. The functional forms presented in the following subsections show which of the formal arguments allowed in the general functional forms given in Table 9.2 are actually used for each fluid.

B.1 Liquid Sodium

The detailed properties for liquid sodium were prepared in connection with the development of COMMIX-1A (pp. 122-123 of Ref. 3) and are based on the information compiled by Golden and Tokar (Ref. 18). Several of the correlations have been implemented in "engineering" units as presented in Ref. 18 and then modified by the appropriate units conversion processes in the coding.

Identification:

$$\text{FLUIDID} (T_{\text{low}}, T_{\text{high}}) = \text{' SODIUM'}$$
(B.1)

$$T_{\text{low}} = 300.0 \quad (\text{B.2})$$

$$T_{\text{high}} = 700.0 \quad (\text{B.3})$$

Density:

$$\begin{aligned} \rho(T) = & 9.50076 * 10^2 - 2.2976 * 10^{-1} * T \\ & - 1.46049 * 10^{-5} * T^2 + 5.63788 * 10^{-9} * T^3 \end{aligned} \quad (\text{B.4})$$

Temperature:

$$\begin{aligned} T(h,P) = & \text{calculated by solving Eq. B.11b for } T \text{ (knowing } h \text{ and } P) \\ & \text{using an iterative numerical procedure} \end{aligned} \quad (\text{B.5})$$

Molecular Viscosity:

$$\mu_t(T) = \frac{3.241903 * 10^{-3} * \exp \left[\frac{508.07}{T + 273.15} \right]}{(T + 273.15)^{0.4925}} \quad (\text{B.6})$$

Molecular Thermal Conductivity:

$$\lambda_t(T) = 92.948 - 5.809 * 10^{-2} * T + 1.1727 * 10^{-5} * T^2 \quad (\text{B.7})$$

Specific Heat Capacity:

$$C_p(T) = 1.43605 * 10^3 - 5.802 * 10^{-1} * T + 4.62506 * 10^{-4} * T^2 \quad (\text{B.8})$$

Derivative of Density with respect to Enthalpy:

$$\begin{aligned} \frac{d\rho}{dh}(T) = & \frac{d\rho}{dT}(T) \text{ from differentiation of Eq. B.4 divided} \\ & \text{by } C_p(T) \text{ from Eq. B.8.} \end{aligned} \quad (\text{B.9})$$

Saturation Pressure:

$$P_{\text{sat}}(T) = \begin{cases} \frac{3.072843 * 10^{11} * \exp \left[\frac{-2.30733 * 10^4}{1.8 * (T + 273.15)} \right]}{[1.8 * (T + 273.15)]^{0.5}} & , T \leq 871.13 \quad (\text{B.10a}) \\ \frac{6.972944 * 10^{11} * \exp \left[\frac{-2.298196 * 10^4}{1.8 * (T + 273.15)} \right]}{[1.8 * (T + 273.15)]^{0.61344}} & , T > 871.13 \quad (\text{B.10b}) \end{cases}$$

Enthalpy:

The enthalpy at any state point (T,P) may be written in terms of the enthalpy at a reference state point (T₀,P₀) as

$$h(T,P) = h(T_0, P_0) + \left[\frac{\partial h}{\partial T} \right]_P * [T-T_0] + \left[\frac{\partial h}{\partial P} \right]_T * [P-P_0] . \quad (\text{B.11a})$$

Taking the reference point to be the saturated liquid line at temperature T₀=T and pressure P₀=P_{sat}(T) and moving along a constant temperature path from P₀ to P, Eq. B.11a may be expressed as

$$h(T,P) = h_{\text{sat}}(T) + \frac{1 + \frac{T}{\rho(T)} * \left[\frac{\partial \rho}{\partial T} \right]_P}{\rho(T)} * [P-P_{\text{sat}}(T)] \quad (\text{B.11b})$$

where (∂ρ/∂T)_P is approximated by dρ/dT (which comes from differentiating Eq. B.4), P_{sat}(T) is given by Eq. B.10, and the saturated liquid enthalpy is given by

$$\begin{aligned}
h_{\text{sat}}(T) = & 2.32444 * 10^3 * \{ -29.02 + 0.389352 * [1.8 * (T+273.15)] \\
& -5.529955 * 10^{-5} * [1.8 * (T+273.15)]^2 \\
& +1.13726 * 10^{-8} * [1.8 * (T+273.15)]^3 \} .
\end{aligned} \tag{B.12}$$

B.2 Liquid Normal Water

The detailed properties for liquid normal water (H₂O) were prepared in connection with the development of COMMIX-1A pp. (123-138 of Ref. 3) and are based on information compiled by Brookhaven National Laboratory. As was the case for the sodium properties in Sec. B.1, several of the correlations have been implemented using "engineering" units forms and then modified by the appropriate units conversion processes in the coding.

Identification:

$$\text{FLUIDID } (T_{\text{low}}, T_{\text{high}}) = ' \quad \text{H2O}' \tag{B.13}$$

$$T_{\text{low}} = 20.0 \tag{B.14}$$

$$T_{\text{high}} = 100.0 \tag{B.15}$$

Density:

$$\rho(h, P) = \begin{cases} \rho_1(h, P) , & h \leq 6.4477 * 10^5 \\ f(y) * \rho_1(h, P) + [1-f(y)] * \rho_2(h, P), & 6.4477 * 10^5 < h \leq 6.57793 * 10^5 \\ \rho_2(h, P) , & h > 6.57793 * 10^5 \end{cases}$$

(B.16a)

(B.16b)

(B.16c)

where

$$\begin{aligned}\rho_1(h,P) = 16.018436 * \{ & [62.4 + 2.14 * 10^{-4} * P_R] \\ & + [- 8.73 * 10^{-5} + 1.438 * 10^{-9} * P_R] * h_R^2 \\ & + [2.32 * 10^{-10} - 6.20 * 10^{-15} * P_R] * h_R^4 \} \end{aligned} \quad (B.17)$$

$$\begin{aligned}\rho_2(h,P) = 16.018436 * \left\{ & [92.924 + 5.761 * 10^{-4} * P_R] \right. \\ & \left. + \frac{[3.94402 * 10^4 + 1.6386 * P_R]}{h_R - [1.37735 * 10^3 + 3.5704 * 10^{-2} * P_R]} \right\} \end{aligned} \quad (B.18)$$

$$f(y) = (8 - 15 * y + 10 * y^3 - 3 * y^5)/16 \quad (B.19)$$

$$y = (h_R - 280)/2.8 \quad (B.20)$$

$$h_R = 4.299226 * 10^{-4} * h \quad (B.21)$$

$$P_R = 1.4503774 * 10^{-4} * P \quad (B.22)$$

Temperature:

$$T(h,P) = \begin{cases} T_1(h,P) , & h < 8.12 * 10^5 \\ f(y) * T_1(h,P) + [1-f(y)] * T_2(h,P), & 8.12 * 10^5 \leq h \leq 8.16 * 10^5 \\ T_2(h,P) , & h > 8.16 * 10^5 \end{cases} \quad \begin{matrix} (B.23a) \\ (B.23b) \\ (B.23c) \end{matrix}$$

where

$$\begin{aligned} T_1(h,P) = & -273.15 + [2.8378 * 10^2 - 2.752333 * 10^{-7} * P] \\ & + [2.4688303 * 10^{-4} + 1.24419 * 10^{-13} * P] * h \\ & + \frac{[1.8796464 * 10^7 - 5.634438 * 10^{-2} * P]}{h - 1.7556418 * 10^6} \end{aligned} \quad (B.24)$$

$$\begin{aligned} T_2(h,P) = & -273.15 + [3.49661 * 10^2 - 2.364921 * 10^{-6} * P] \\ & + [1.1964506 * 10^{-5} + 6.291758 * 10^{-12} * P] * h \\ & + [2.294645 * 10^{-10} - 5.990103 * 10^{-18} * P] * h^2 \\ & + [-8.587812 * 10^{-17} + 2.015452 * 10^{-24} * P] * h^3 \end{aligned} \quad (B.25)$$

and $f(y)$ and y are given by Eqs. B.19 and B.20, respectively.

Molecular Viscosity:

$$\mu_t(h,P) = \left\{ \begin{array}{l} 1.2994702 * 10^{-3} - 9.2640321 * 10^{-4} * \chi \\ + 3.8104706 * 10^{-4} * \chi^2 \\ - 8.2194445 * 10^{-5} * \chi^3 + 7.022438 * \chi^4 \\ - (-6.5959 * 10^{-12} + 6.763 * 10^{-12} * \eta + 2.88825 * 10^{-12} * \eta^2 \\ + 4.4525 * 10^{-13} * \eta^3) * (P - 6.8945753 * 10^5), \\ h \leq 2.76 * 10^5 \end{array} \right. \quad (B.26a)$$

$$\mu_t(h,P) = \left\{ \begin{array}{l} 1.4526053 * 10^{-3} - 6.9880085 * 10^{-9} * h \\ + 1.5210230 * 10^{-14} * h^2 - 1.2303195 * 10^{-20} * h^3 \\ + (-3.8063508 * 10^{-11} + 3.9285208 * 10^{-16} * h \\ - 1.2585799 * 10^{-21} * h^2 + 1.2860181 * 10^{-27} * h^3) \\ * (P - 6.8945753 * 10^5), \\ 2.76 * 10^5 \leq h \leq 3.94 * 10^5 \end{array} \right. \quad (B.26b)$$

$$\left\{ \begin{array}{l} 3.0260323 * 10^{-4} - 1.8366069 * 10^{-4} * \zeta \\ + 7.5670758 * 10^{-5} * \zeta^2 - 1.6478789 * 10^{-5} * \zeta^3 \\ + 1.4164576 * 10^{-6} * \zeta^4, \\ h > 3.94 * 10^5 \end{array} \right. \quad (B.26c)$$

where

$$\chi = (h - 4.265884 * 10^4) / 1.165326 * 10^5 \quad (B.27)$$

$$\eta = (h - 5.53588 * 10^4) / 1.542138 * 10^5 \quad (B.28)$$

$$\zeta = (h - 4.014676 * 10^4) / 2.5695322 * 10^5 \quad (B.29)$$

Molecular Thermal Conductivity:

$$\begin{aligned} \lambda_t(h) = & 0.57373862 + 0.25361036 * \left[\frac{h}{5.815 * 10^5} \right] \\ & - 0.14546827 * \left[\frac{h}{5.815 * 10^5} \right]^2 + 0.013874725 * \left[\frac{h}{5.815 * 10^5} \right]^3 \end{aligned} \quad (B.30)$$

Specific Heat Capacity:

$$C_p(h,P) = \begin{cases} C_{P,1}(h,P) , & h \leq 8.12 * 10^5 \\ f(x) * C_{P,1}(h,P) + [1-f(x)] * C_{P,2}(h,P) , & 8.12 * 10^5 < h \leq 8.16 * 10^5 \\ C_{P,2}(h,P) , & h > 8.16 * 10^5 \end{cases} \quad \begin{matrix} (B.31a) \\ (B.31b) \\ (B.31c) \end{matrix}$$

where

$$C_{P,1}(h,P) = \left\{ [2.4688303 * 10^{-4} + 1.24419 * 10^{-13} * P] - \frac{1.8790464 * 10^7 - 5.634438 * 10^{-2} * P}{[h - 1.7556418 * 10^{-6}]^2} \right\}^{-1} \quad (B.32)$$

$$\begin{aligned} C_{P,2}(h,P) = & \{ [1.1964506 * 10^{-5} + 6.291758 * 10^{-12} * P] \\ & + [4.58929 * 10^{-10} - 1.1980206 * 10^{-17} * P] * h \\ & + [-2.576343 * 10^{-16} + 6.046356 * 10^{-24} * P] * h^2 \}^{-1} \end{aligned} \quad (B.33)$$

$$x = (h - 8.14 * 10^4)/2000 \quad (B.34)$$

and $f(x)$ is evaluated using Eq. B.19.

Derivative of Density with respect to Enthalpy:

$$\frac{d\rho}{dh}(h,P) = \text{partial derivative of Eq. B.16 with respect to } h \text{ at constant } P. \quad (\text{B.35})$$

Saturation Pressure:

$$P_{\text{sat}}(T) = \text{calculated by solving the following saturation temperature function for } P \text{ (knowing } T) \text{ using an iterative numerical procedure.} \quad (\text{B.36})$$

$$T_{\text{sat}}(P) = \begin{cases} T_{\text{sat},1}(P) , & P_R \leq 43.4302 & (\text{B.37a}) \\ f(a) * T_{\text{sat},1}(P) + [1-f(a)] * T_{\text{sat},2}(P) , & 43.4302 < P_R \leq 45.4298 & (\text{B.37b}) \\ T_{\text{sat},2}(P) , & 45.4298 < P_R \leq 1069.2 & (\text{B.37c}) \\ f(b) * T_{\text{sat},2}(P) + [1-f(b)] * T_{\text{sat},3}(P) , & 1069.2 < P_R \leq 1090.8 & (\text{B.37d}) \\ T_{\text{sat},3}(P) , & P_R > 1090.8 & (\text{B.37e}) \end{cases}$$

where

$$T_{\text{sat},1}(P) = \left[\frac{2.6347 * 10^3}{6.026 - P_L} - 3.67486 * 10^2 + 4.484 * P_L \right] / 1.8 \quad (\text{B.38})$$

$$T_{\text{sat},2}(P) = \left(73.802 + 65.14 * P_L + 24.859 * P_L^2 - 4.3391 * P_L^3 + 1.6889 * P_L^4 \right) / 1.8 \quad (\text{B.39})$$

$$T_{\text{sat},3}(P) = \left[5.88994 * 10^2 + 5.5386 * 10^{-2} * P_R - 3.516 * 10^{-6} * P_R^2 - \frac{2.26805 * 10^5}{P_R + 7.6885 * 10^2} \right] / 1.8 \quad (\text{B.40})$$

$$a = (P_R - 44.98) / 0.4498 \quad (\text{B.41})$$

$$b = (P_R - 1080) / 10.80 \quad (\text{B.42})$$

$$P_R = 1.4503774 * 10^{-4} * P \quad (\text{B.43})$$

$$P_L = \log_{10}(P_R) \quad (\text{B.44})$$

and $f(a)$ and $f(b)$ are evaluated using Eq. B.19.

Enthalpy:

$$h(T,P) = \text{calculated by solving Eq. B.23 for } h \text{ (knowing } T \text{ and } P) \text{ using an iterative numerical procedure.} \quad (\text{B.45})$$

B.3 Liquid Heavy Water

The detailed properties for liquid heavy water (D_2O) are new with this release of COMMIX-1AR/P and are based on information compiled by Lo (Ref. 19) and Dunn (Ref. 20). These properties are intended for use in the pressure range from 1 kPa to 2 MPa; however, there

is no pressure dependence in any of the functional forms. (The functional forms used in these properties was constrained by what is allowed in the SAS4A/SASSYS computer programs.) The symbol T_K is the temperature in units of kelvins [K], or

$$T_K = T + 273.15 \quad (\text{B.46})$$

on the right-hand side of all of the following functional forms.

Identification:

$$\text{FLUDID } (T_{\text{low}}, T_{\text{high}}) = ' \quad \text{D2O}' \quad (\text{B.47})$$

$$T_{\text{low}} = 0.0 \quad (\text{B.48})$$

$$T_{\text{high}} = 200.0 \quad (\text{B.49})$$

Density:

$$\rho(T) = 9.1568 * 10^2 + 1.5447 * T_K - 3.0790 * 10^{-3} * T_K^2 \quad (\text{B.50})$$

Temperature:

$$\begin{aligned} T(h) = & \text{calculated by solving Eq. B.56 for } T \text{ (knowing } h) \\ & \text{using an iterative numerical procedure.} \end{aligned} \quad (\text{B.51})$$

Molecular Viscosity:

$$\begin{aligned} \mu_t(T) = & - 5.8232 * 10^{-3} + 7.9819161 * T_K^{-1} \\ & - 3.5853469 * 10^3 * T_K^{-2} + 5.4455172 * 10^5 * T_K^{-3} \end{aligned} \quad (\text{B.52})$$

Molecular Thermal Conductivity:

$$\begin{aligned} \lambda_t = & - 0.28088 + 5.0999 * 10^{-3} * T_K \\ & - 8.0711 * 10^{-6} * T_K^2 + 2.5964 * 10^{-9} * T_K^3 \end{aligned} \quad (\text{B.53})$$

Specific Heat Capacity:

$$C_p(T) = 5.1937 * 10^3 - 7.1933 * (644.5 - T_K) + 1.2557 * 10^{-2} * (644.5 - T_K)^2 \quad (B.54)$$

Derivative of Density with respect to Enthalpy:

$$\frac{d\rho}{dh}(T) = \frac{d\rho}{dT}(T) \text{ from differentiation of Eq. B.49 divided by } C_p(T) \text{ from Eq. B.53.} \quad (B.55)$$

Saturation Pressure:

$$P_{sat}(T) = \exp(2.2979 * 10^1 - 3.5041 * 10^3 * T_K^{-1} - 2.9432 * 10^5 * T_K^{-2}) \quad (B.56)$$

Enthalpy:

The enthalpy is determined by integrating the specific heat capacity over temperature, using a reference point of $h=0$ at $T_K = 0$ K, which may be expressed as

$$h(T) = \int_0^{T_K} C_p(T_K) dT_K \quad (B.57)$$

Using the functional form for C_p given by Eq. B.54, this integration is performed analytically to obtain the functional form implemented in the coding.

B.4 Helium Gas

The detailed properties for helium gas are new with this release of COMMIX-1AR/P and are based on the information compiled by Vargaftik (Ref. 21) and Raznjevic (Ref. 22). The following functional forms are intended to be used in the temperature range covering 200°C to 800°C and the pressure range covering 0.1 MPa to 20 MPa.

Identification:

$$\text{FLUDID } (T_{\text{low}}, T_{\text{high}}) = \text{HE}' \quad (\text{B.58})$$

$$T_{\text{low}} = 0.0 \quad (\text{B.59})$$

$$T_{\text{high}} = 1000.0 \quad (\text{B.60})$$

Density:

$$\rho(T, P) = \frac{P}{Z(T, P) * 2076.69 * (T + 273.15)} \quad (\text{B.61})$$

where

$$Z(T, P) = \begin{cases} 1.0, & P \leq 10^5 \\ 1.0 + P * \left[-3.90 * 10^{-10} + \frac{1.523 * 10^{-6}}{T + 273.15} \right], & P > 10^5 \end{cases} \quad (\text{B.62a})$$

$$(\text{B.62b})$$

Temperature:

$$T(h) = \text{calculated using functional form obtained by analytically solving Eq. B.69b for } T. \quad (\text{B.63})$$

Molecular Viscosity:

$$\mu_t(T) = 1.9104 * 10^{-5} + 4.1255 * 10^{-8} * T - 8.7633 * 10^{-12} * T^2 \quad (\text{B.64})$$

Molecular Thermal Conductivity:

$$\lambda_t(T) = 0.14312 + 3.5203 * 10^{-4} * T - 8.5715 * 10^{-8} * T^2 \quad (\text{B.65})$$

Specific Heat Capacity:

$$C_p = 5192. \quad (B.66)$$

Derivative of Density with respect to Enthalpy:

$$\frac{d\rho}{dh}(T,P) = \text{partial derivative of Eq. B.57 with respect to temperature} \quad (B.67)$$

at constant pressure divided by C_p from Eq. B.66.

Saturation Pressure:

The saturation pressure function is only used in COMMIX to check for the attainment of boiling conditions for fluid number 1. Since helium is always considered to be a gas, this check should not be enabled by the user (i.e. take the default value of 0 for input variable IBOIL) when using the detailed helium properties. The coding contains

$$P_{sat} = 10^{-5} \quad (B.68)$$

since the FUNCTION must exist to avoid an error message when forming an executable binary version of COMMIX.

Enthalpy:

The enthalpy is determined by integrating the specific heat capacity over temperature, using a reference point of $h=0$ at $T = -273.15^\circ\text{C}$, which may be expressed as

$$h(T) = \int_{-273.15}^T C_p dT \quad (B.69a)$$

Using the functional form for C_p given by Eq. B.66, this integration is performed analytically to obtain

$$h(T) = C_p * (T + 273.15) \quad (B.69b)$$

as the functional form implemented in the coding.

APPENDIX C: INSTALLATION AND EXECUTION INSTRUCTIONS

The recommended procedure is for one person at a site to be the "owner" of the code system. The owner installs the code system and informs the other users of procedures for executing the code, passing along sample execution jobstreams, input files, the input guide, and other forms of documentation. Under the recommended installation procedure, the owner compiles the several code sections and creates a series of object libraries, which are then made accessible to the other users for execution. Under this procedure, the general user does not need access to the source code. The installation procedure is described in Sec. C.1; the general user need only skim through this section. The execution procedure is described in Sec. C.2; this information is important to all users.

The COMMIX computer program system has been successfully installed and executed on both mainframe and workstation computers made by a variety of vendors, including Cray Research Incorporated, Digital Equipment Corporation, Hewlett Packard, International Business Machines, and Sun Microsystems. The installation and execution instructions can not be shown for all machine types in an appendix such as this; instead, examples are shown here for two computing environments and additional specific information will be included as a part of the code transmittal to your site. The examples shown here are based on the two mainframe batch computing environments under which COMMIX is used at Argonne National Laboratory (ANL): a Cray X-MP running UNICOS with NQS and an IBM 3084 running MVS/JES3. Additionally, the Cray examples assume operation under the Bourne shell. These examples will need a certain amount of modification for use at other sites; although these modifications are discussed as part of the installation instructions in Sec. C.1, the person installing the code will need a good working knowledge of their computer system in order to make the appropriate changes.

C.1 Installation

Transmittal of the COMMIX code system consists of a number of files, typically, the base COMMIX code, several sets of detailed fluid properties, four graphics postprocessor codes, an input guide, and several sample problems; the specific files, their format, and their order will

be noted as part of the transmittal. The source code you receive is for a specific machine type; the primary differences between the code versions are whether the machine has (a) a short or long word length and (b) vector hardware. The work needed to install the code system on your machine is described in this section.

C.1.1 Background

Several items relating to code structure are important to an understanding of the recommended installation procedure: data storage allocation, detailed fluid properties, and special system routines. These are discussed in this subsection; specific installation instructions are described in Sec. C.1.2.

A quasidynamic storage allocation method is used in COMMIX and the four graphics postprocessor codes. Most arrays are allocated dynamically during execution in COMMON BLOCK /SPACE/ to the size needed for the case. The user must, however, specify how much storage is to be made available in this COMMON BLOCK. This is accomplished by specifying the dimension of array S in COMMON BLOCK /SPACE/ in SUBROUTINE ALTER; this is the only routine which needs the true dimension of array S. During input processing, the code will compute and print the minimum dimension of array S required for execution of the case; if the required dimension is larger than the actual dimension, the code will terminate with an error message saying that the user should increase the dimension of array S and resubmit the job. By adjusting the dimension of array S to the minimum size needed for a specific case, the user can minimize the execution memory requirement, which is a factor in job turnaround time and cost on many systems. For this reason (and the reason in the second paragraph below), the recommended procedure is to compile the base COMMIX code and the four graphics postprocessor codes to create a set of files containing object code; the user then compiles SUBROUTINE ALTER and links this to the rest of the appropriate object code for execution. The person installing COMMIX need not be concerned with the other details of how arrays are actually allocated within COMMON BLOCK /SPACE/; these other details are described in the Programmer's Guide (Ref. 14).

There are four instances in the base COMMIX code where provision has been made for fixed storage limitations to be overridden. These storage limits are set using PARAMETER statements in the source code. The person installing the code system may wish to alter these limits, by changing the PARAMETER statements throughout the base COMMIX code, before compilation. Any changes made to these PARAMETER values should be communicated to the code users at your site. The four instances are as follows, showing the applicable variable in the PARAMETER statement, the value set in the source code as distributed, limitations on the value to be used, and the section of this report which discusses the use of this variable. In each instance, the variable in the PARAMETER statement is of type INTEGER and the minimum allowed value is 1. (1) The maximum number of thermal structure prototypes to be allowed in any calculation is set by PNSTRU; the value set in the source code is 99; the storage algorithm requires that this value not exceed 9999 (however, some FORMAT statements only allow a 3-digit output field width); thermal structures are discussed in Sec. 5. (2) The maximum number of radiation surfaces to be allowed in any calculation is set by PNRAD; the value set in the source code is 99; there is no useful purpose served by this value being greater than $2 \times \text{PNSTRU}$ (however, some FORMAT statements only allow a 3-digit output field width); radiation surfaces are discussed in Sec. 5 as part of the thermal structure model. (3) The maximum number of nonadjacent cells coupled to a given cell by thermal structures when using the full-coupling numerics ($\text{IHTWALL}(99)=2$) in any calculation is set by NTSCUP and NODD; both variables must be set to the same value; the value set in the source code is 2; there is no upper limit for this value; cell-to-cell coupling limitations are discussed in Sec. 5.4 as part of the thermal structure model. (4) The maximum number of fluids to be allowed in any calculation is set by PIFHTX; the value set in the source code is 5; there is no upper limit imposed (however, some FORMAT statements only allow a 1-digit output field width); the use of multiple fluids is discussed in Secs. 3.3 and 9.1.

Fluid number 1 in a COMMIX case may use either detailed or simplified properties; all other fluids must use simplified properties (cf. Sec. 9.1); therefore, only one set of detailed properties may be used in any given case. Four sets of detailed properties are distributed with the COMMIX code system: liquid sodium, liquid normal water, liquid heavy water, and helium gas. Only one of these sets may be included when forming an executable version of COMMIX.

In order to allow the user the flexibility to use any one of the detailed properties sets, the recommended procedure is for a separate file to be created for each set of fluid properties which contains the object code for that set of properties; the user then links one of these properties files with the base COMMIX object code and system routines to form a complete executable COMMIX version when running a case.

The COMMIX code system needs various system routines which are not necessarily part of the normal machine environment to be executable: LOCF, TLEFT, LINPACK, DISSPLA, DATE1, TIME1, and ANLACT. These routines or packages of routines are described below. Table C.1 provides a summary showing which of these routines are needed by the various code sections and the file names used in the example jobstreams to reference these routines. If these routines are not available on your system, the source code will need to be modified and/or replacement routines written at your installation. Even if the routines exist, you will probably need to modify the execution example jobstreams to refer to the proper object files in use at your installation.

Table C.1 Special System Routines Needed and Locations
Assumed in Sample Jobstreams

Routine or Package Needed	Needed By					IBM Location	Cray Location
	B A S E	V E C T	T E M P	T D S C	G R P H		
LOCF	x	x	x	x	x	SYS1.AMDLIB	Use LOC, an inline intrinsic.
TLEFT	x					SYS1.AMDLIB	\$ANLUTIL
LINPACK	x					SYS1.AMDLIBX	/lib/libsci.a
DISSPLA		x	x		x	CA.DIS110.DISLIB.LOAD CA.DIS110.DRVS.DYNLIB.LOAD CA.DIS110.DRVS.GRAFPROD.SITEDATA	\$DISLIB11
DATE1		x	x			SYS1.AMDLIB	Use DATE from /lib/lib*.a
TIME1		x	x			SYS1.AMDLIB	Use CLOCK from /lib/lib*.a
ANLACT		x	x			SYS1.AMDLIB	- not used -

A function named LOCF, of type INTEGER having one argument, is used in connection with data storage. The result of LOCF(var) after dividing by NOBPI is the absolute core storage address in computer words of the variable "var". This function is called by SUBROUTINE ALTER in the base COMMIX code and all four graphics postprocessors; it is additionally called by PROGRAM COMMIX and SUBROUTINES CLEAR and RESTAR in the base COMMIX code. On the IBM at ANL, LOCF is a locally written system function whose object code is in SYS1.AMDLIB and returns a value in bytes; the IBM version of the COMMIX code system has NOBPI=4 to convert the result to words; the person installing the COMMIX code system will have to make modifications to simulate the behavior of this routine on other IBM systems. The Cray version of the COMMIX code system uses standard system function LOC instead of LOCF and, since LOC returns the address in words, has NOBPI=1; LOC is an inline intrinsic function in cft77; therefore, no special action should be required by the person installing the code system to access this function on other Cray systems.

The base COMMIX code periodically calls FUNCTION TLEFTS, having one dummy argument, which returns the amount of Central Processor Unit (CPU) time, in seconds, available for the job to complete; this value is used in conjunction with input variable TREST in NAMELIST DATA for an orderly termination of the case before getting a "CPU Time Limit" system error message (cf. Sec. 12.1). The source code for FUNCTION TLEFTS is included in the base COMMIX code file; this function, in turn, calls a system-specific function, assumed to have the name TLEFT, which returns the time remaining. On the Cray and IBM computers at ANL, the CPU time remaining, in hundredths of CPU seconds, is returned by a locally written system function named TLEFT; the object code for TLEFT is in the library pointed to by shell variable ANLUTIL (defined by /etc/profile) on the Cray and in SYS1.AMDLIB on the IBM. The person installing the COMMIX code system will need to modify FUNCTION TLEFTS to call the appropriate routine and modify the libraries referenced in the execution examples to point to the appropriate object file for your system. The foregoing assumes that COMMIX will be executed as a batch job, which is the recommended operating mode. For execution in interactive mode or on systems where the equivalent of TLEFT cannot be simulated, FUNCTION TLEFTS should be modified to return a value greater than 10.0; an example of such a function is shown in Table C.2. The other users should be informed of any

changes to the functionality of TLEFTS, since the code will stop if the value returned by TLEFTS is less than the value input by the user for variable TREST in NAMELIST DATA.

Table C.2 FORTRAN-77 Example of FUNCTION TLEFTS for Use
in Interactive Mode¹

```
FUNCTION TLEFTS(TDUM)
REAL TLEFT,TDUM
TLEFTS=3600.0
RETURN
END
```

Notes:

1. This replacement will also be useful on systems which do not have a function which returns "CPU time remaining in job".
2. The value returned via TLEFTS must be greater than 10.0 (and greater than the value input for TREST in NAMELIST DATA).
3. Replace "REAL" with "DOUBLE PRECISION" on short-word machines.

The base COMMIX code calls various LINPACK (Ref. 23) routines to solve the system of linear equations for radiation heat transfer (i.e. Eq. 6.39 in Ref. 1). The routines called directly by COMMIX (from SUBROUTINE QSRAD) are SGEFA and SGESL (for single-precision variables) on long-word-length machines (e.g. Cray) and DGEFA and DGESL (for double-precision variables) on short-word-length machines (e.g. IBM). On the Cray, LINPACK is in one of the default system libraries (/lib/libsci.a) used by the segment loader; therefore, no special action should be needed by the person installing COMMIX to access this package on the Cray. On the IBM system at ANL, the object code for LINPACK is in SYS1.AMDLIBX; the person installing COMMIX will need to need modify the execution examples to point to the appropriate object file for your IBM system (and/or revise QSRAD to call other routines if LINPACK is not available).

Postprocessor codes VECT, TEMP, and GRPH produce graphical output by making calls to routines in the DISSPLA (Ref. 24) package (specifically, the FORTRAN-77 library at Release

11.0). If DISSPLA is not available at your installation, there is no point in installing these three codes or TDSC (which prepares input for GRPH) and you may ignore the rest of this paragraph. On the Cray at ANL, shell variable DISLIB11 (defined by /etc/profile) contains the complete name of the DISSPLA object library; the person installing COMMIX will need to define this shell variable (or replace its usage in the segldr invocation) in the execution example jobstreams for distribution to other users. On the IBM at ANL, the main object code for DISSPLA is referenced through the PRELIB parameter in the cataloged procedure FORTVCEG; additionally, DDDNAMEs GO.CADYNLIB and GO.FT44F001 reference DISSPLA system files in the execution jobstreams; the person installing COMMIX will need to make the appropriate changes in the execution jobstreams for distribution to the other users. Codes VECT, TEMP, and GRPH produce graphics in device-independent DISSPOP metafile format (via CALLing COMPRs); the resulting file must be further postprocessed to produce viewable output on an actual graphical device (e.g. video terminal, plotter, film); the example jobstreams move the DISSPOP metafile to a PostScript-type laser printer using a locally written system script named hardcopy on the Cray and a locally written cataloged (system) procedure named PPS on the IBM; the person installing COMMIX may need to modify this mode of operation for your system; this may require source code changes and will require modification of the execution example jobstreams before distributing them to the other users.

Graphics postprocessor codes VECT and TEMP call up to three routines to obtain information (job name, job number, user name, date, and time) to be used for an extra figure label section; this extra figure label was added since the border label drawn by DISSPLA fell outside of the area printable on a laser-writer output device. On the Cray, the job name, job number, and user name are obtained from standard system-defined shell variables QSUB_REQNAME, QSUB_REQID, and LOGNAME and the date and time are obtained from standard system routines DATE and CLOCK, which are in one of the default libraries (/lib/lib*.a) used by the segment loader; no special action should be needed by the person installing COMMIX to access this information. On the IBM at ANL, this information is obtained from locally written system routines ANLACT, DATE1, and TIME1, whose object code is in SYS1.AMDLIB; the person installing the COMMIX code system will need to modify the source code (SUBROUTINE VDRIVE in VECT and SUBROUTINE IDRIVE in TEMP) to

call other routines (or delete the sections of coding referencing these routines and their results) and modify the libraries referenced in the example jobstreams for use on other IBM systems.

C.1.2 Specific Instructions

The code sections to be installed are listed in Table C.3. Each code section shown in this table will be an individually accessible component of the code transmission to your site. At a minimum, you must install at least two of these code sections: (1) the base COMMIX code and (2) one set of detailed fluid properties; you may wish to also install one or more of the other sets of detailed fluid properties (depending on the types of cases to be run by your users) and the four graphics postprocessor codes (depending on the availability of DISSPLA and other system capabilities).

Table C.3 Code Sections to Be Installed¹

Name ²	Description
BASE	COMMIX without any detailed fluid properties
PROPNA	detailed sodium properties
PROPH2O	detailed normal water properties
PROPD2O	detailed heavy water properties
PROPHE	detailed helium properties
VECT	graphics postprocessor "VECT"
TEMP	graphics postprocessor "TEMP"
TDSC	graphics postprocessor "TDSC"
GRPH	graphics postprocessor "GRPH"

Notes:

1. As a minimum, the "base" COMMIX code and one set of "detailed" fluid properties must be installed.
2. The code section name is represented by the string "xxxx" in the example installation jobstreams (Tables C.4. and C.9).

Installation consists of three steps: (1) creation of an object code library for each code section, (2) revising the execution example jobstreams to reference these and various system-specific libraries, and (3) making the revised execution example jobstreams and other information available to the other users at your site.

Object code libraries are created by compiling the source code for each code section to be installed. Examples are discussed below for Cray and IBM mainframe systems. The examples assume that the source code is available on disk; the procedures for establishing the source code files on disk will be covered as part of the transmittal, since the procedures will vary depending on the mode of transmittal. The examples also assume that you have made any source code changes necessary to accommodate the system-dependent routines noted in Table C.1, as discussed in Sec. C.1.1.

C.1.2.1 Cray

A sample jobstream to compile and create an object library on the Cray is shown in Table C.4. This job assumes that the source code for code section "xxxx" is in file `$CMX_DIR/SOURCE/xxxx`, where `CMX_DIR` is a shell variable to be defined by you in the job and "xxxx" is one of the code section names shown in Table C.3. This job will leave the compilation listing on file `$CMX_DIR/LIST/xxxx` and the object library on file `$CMX_DIR/OBJECT/xxxx`. (This job will create the LIST and OBJECT subdirectories if they do not already exist.) You may need to alter the line "cd \$TMPDIR" in the section labeled "Prologue" of this job; the intent of this command is to use a unique directory during this job to hold all temporary files (which are created and later deleted by the job); at ANL, system-defined shell variable `TMPDIR` points to such a directory and the directory is automatically removed at the end of the job. Compilation is done using "cft77"; the object code is then passed to "bld" to form an object code library. The final section of the job adds the compilation listing to "stdout"; you may want to delete this or replace it with other UNICOS commands (e.g. `lpr`, `rcp`, or `dispose`) depending on your system configuration. This job should be submitted for each code section to be installed (changing "xxxx" to the name of the code section).

Sample jobstreams for executing COMMIX and the graphics postprocessors are shown in Tables C.5 through C.8. You need to modify these before distributing them to the other users at your site. A number of these changes are common to all these samples. All of these jobs use a temporary unique directory to hold all temporary files for the job; see discussion of "cd \$TMPDIR" in the above paragraph. You may want to alter the way in which printable output

having carriage-control information in column 1 is handled; in the examples, this output is collected on file \$HOME/thisjob.l during the job (where the user is assumed to have changed the string "thisjob" to the actual job name prior to submittal) and added to stdout at the end of the job. The binary restart and plot files created by COMMIX are assumed to be left on the Cray disks as files \$HOME/thisjob.r and \$HOME/thisjob.p, respectively; you may want to recommend alternate permanent file names and storage locations for your users. You may want to adjust the "QSUB" parameters for your system (e.g., add the "-q queue_name" option). Finally, depending on how your users submit jobs to the Cray, you may need to surround the samples shown with other control statements and add additional commands in the Prologue and Epilogue sections. The additional changes needed will be discussed for each type of job.

The sample jobstream for executing COMMIX is shown in Table C.5. In the Prologue section, you need to alter the definition of two shell variables: CMXBASE and CMXPROP. Shell variable CMXBASE is the name of the object library containing the base COMMIX code; it should be specified completely. Shell variable CMXPROP is the name of the object library containing the one set of detailed fluid properties to be used for a case; you should adjust the initial part of this file name to conform to how you installed the object code sections and modify the comment in this section to indicate the available choices to the user if all four sets of detailed fluid properties were not installed, leaving it to the user to select one set of properties by completing the file name when running a case. In the "segldr" invocation, replace the use of shell variable ANLUTIL with the name of the file that contains the object code for the function called from TLEFTS at your site which replaces ANL function TLEFT.

The sample jobstream for executing graphics postprocessor codes VECT and TEMP is shown in Table C.6. In the Prologue section, you need to alter the definition of shell variable CMXPLOT; however, leave this definition such that the user changes "xxxx" to either "VECT" or "TEMP" to select which postprocessor code to run. In the "segldr" command, replace the use of shell variable DISLIB11 with the name of the file containing the object library of DISSPLA. In the Epilogue section, change the use of the "hardcopy" script to the appropriate command(s) needed to send the graphics to a real device at your installation.

The sample jobstream for executing graphics postprocessor code TDSC is shown in Table C.7. In the third line of the "segldr" command, supply the complete name of the file which contains the object library for TDSC.

The sample jobstream for executing graphics postprocessor code GRPH is shown in Table C.8. Two changes are needed in the "segldr" command. (1) In the first line, replace the use of shell variable DISLIB11 with the name of the file containing the object library of DISSPLA. (2) In the third line, supply the complete name of the file which contains the object library for GRPH. In the Epilogue section, change the use of the "hardcopy" script to the appropriate command(s) needed to send the graphics to a real device at your installation.

Table C.4 Cray Batch Job to Compile Code Section

```
# user=
#
# QSUB  -r COMPILE  -lT 500  -lM 1500Kw  -eo
#      jobname      time      memory      stderr
#
# Activate full command echo with timestamp.
set -vxS
#
# #####
# COMMIX: Compile code module 'xxxx' and save in object library.
# -----
# Before submitting this job:
#   - Change 'xxxx' to the name of the appropriate code module.
#   - Insure shell variable 'TMPDIR' is a valid directory.
#   - Define shell variable 'CMX_DIR' in 'Prologue' section.
# -----
# Assumes running under UNICOS (Bourne shell) as NQS batch job.
# Permanent files used by this job:
#   $CMX_DIR/SOURCE/xxxx: source code (input to cft77).
#   $CMX_DIR/LIST/xxxx: compiler listing (output from cft77).
#   $CMX_DIR/OBJECT/xxxx: object library (output from bld).
# Shell variable 'CMX_DIR' pointing to COMMIX directory must be
# defined by user in 'Prologue' section below. Subdirectories
# '$CMX_DIR/LIST' and '$CMX_DIR/OBJECT' are created by this job
# in 'Prologue' section if they do not already exist.
# Temporary files created and deleted by this job are all in the
# directory set in 'Prologue' section using shell variable 'TMPDIR':
#   fort.f, fort.l, fort.o, NBL.
# #####
#
# #####
# Prologue: Directory Definitions
# #####
#
# Set current working directory for this job to hold all temporary
# files. Assumes that shell variable 'TMPDIR' exists and is the
# complete path name of this directory, the directory exists, and
# you have full access to it.
```

Table C.4 Cray Batch Job to Compile Code Section (Cont'd)

```

cd $TMPDIR
#
# Define shell variable 'CMX_DIR' pointing to COMMIX directory.
# This directory must already exist and you must have full access.
CMX_DIR=$HOME/COMMIX1ARP
#
# Create subdirectories to hold compiler listing and object library
# if they do not already exist.
test -d $CMX_DIR/LIST || mkdir -m 700 $CMX_DIR/LIST
test -d $CMX_DIR/OBJECT || mkdir -m 755 $CMX_DIR/OBJECT
#
# #####
# Compile source code from file '$CMX_DIR/SOURCE/xxxx' producing
# a listing on file '$CMX_DIR/LIST/xxxx' and object code on file
# './fort.o'. Terminate job if compilation errors.
# #####
cp $CMX_DIR/SOURCE/xxxx fort.f
cft77 -e isz -d px -m 1 -V -- fort.f
ERRCFT=$?
cp fort.l $CMX_DIR/LIST/xxxx
chmod 600 $CMX_DIR/LIST/xxxx; ls -l $CMX_DIR/LIST
rm fort.f fort.l
if test $ERRCFT -ne 0; then banner 'ERR-cft'; rm fort.o; exit; fi
#
# #####
# Using object code on file './fort.o', generate object library on
# file './NBL'; if library generation is successful, copy library
# to permanent file '$CMX_DIR/OBJECT/xxxx' and mark readable.
# #####
bld rv NBL fort.o
if test $? -ne 0; then banner 'ERR-bld'; rm fort.o NBL; exit; fi
cp NBL $CMX_DIR/OBJECT/rxxx
chmod 644 $CMX_DIR/OBJECT/xxxx; ls -l $CMX_DIR/OBJECT
rm fort.o NBL
#
# #####
# Add compilation listing to 'stdout' (col. 1 = CC). (OPTIONAL)
# #####
echo '1' >fort.f; cat $CMX_DIR/LIST/xxxx >>fort.f; echo '1' >>fort.f
set +vxS; asa fort.f; set -vxS
rm fort.f
#

```

Table C.5 Cray Batch Job to Execute COMMIX

```

# user=
#
# QSUB -r thisjob -lT 3600 -lM 800Kw -eo
#      jobname      time      memory      stderr
#
# Activate full command echo with timestamp.
set -vxS
#
# #####
# COMMIX: Execution.
# -----
# Before distributing this jobstream to users, installer should:
#   - In 'Prologue': Adjust definitions of shell variables CMXBASE
#     and CMXPROP.
#   - In 'segldr' call: Replace use of shell variable ANLUTIL with
#     name of file containing object code for TLEFT (and anything
#     else special needed for your system).
# Before submitting this job, user should:
#   - Change 'thisjob' to an actual jobname.
#   - In 'Prologue' section: insure shell variables 'TMPDIR' and
#     'CMXPROP' are properly set and used; properly define files
#     './SOURCE', './CARDIN', and './CMXOLD'.
#   - In 'Epilogue' section: adjust file dispositions.
# -----
# Assumes running under UNICOS (Bourne shell) as NQS batch job.
# Permanent COMMIX system object files read by this job:
#   $CMXBASE: base COMMIX code object library.
#   $CMXPROP: detailed fluid properties object library.
# Shell variables 'CMXBASE' and 'CMXPROP' pointing to COMMIX files
# are defined in 'Prologue' section below.
# Permanent user files used by this job:
#   $HOME/priorjob.r: binary restart file from prior job (input).
#   $HOME/thisjob.l: printable output with carriage control.
#   $HOME/thisjob.r: binary restart file created by this job.
#   $HOME/thisjob.p: binary plot file created by this job.
# Temporary files created and deleted by this job are all in the
# directory set in 'Prologue' section using shell variable 'TMPDIR':
#   CARDIN, CMXNEW, CMXOLD, COMMIX.a, fort.f, fort.l, fort.o,
#   LDRERR, LDROUT, PLTDAT, SOURCE.
# #####
#
# #####
# Prologue: Directory and File Definitions
# #####
#
# Set current working directory for this job to hold all temporary
# files. Assumes that shell variable 'TMPDIR' exists and is the
# complete path name of this directory, the directory exists, and
# you have full access to it.
cd $TMPDIR
#
# Define shell variables pointing to COMMIX system object files;
# definitions depend on how code was installed on your system.
# 'CMXBASE': base code object library.
CMXBASE='/owner_id_info/COMMIX1ARP/OBJECT/BASE'
# 'CMXPROP': object library containing detailed properties for
# fluid number 1; change 'xxxx' to 'NA' for sodium, 'H2O' for normal
# water, 'D2O' for heavy water, and 'HE' for helium.
CMXPROP='/owner_id_info/COMMIX1ARP/OBJECT/PROPxxxx'
#

```


Table C.5 Cray Batch Job to Execute COMMIX (Cont'd)

```
# File './SOURCE' contains routines to be compiled. Subroutine
# ALTER must always be present; adjust dimension of array S for size
# of problem. Add functions HEATCU, CPSOL1, and XKSOL1 if needed.
set +v
cat <<'EOF1' >SOURCE
    SUBROUTINE ALTER(NAVAIL)
    INTEGER S,SPAEND
    COMMON /SPACE/ S(221000),SPAEND
    COMMON /MACHIN/ NOBPI,NOWPD,NOWPS
    NAVAIL=(LOCF(SPAEND)-LOCF(S(1)))/NOBPI
    RETURN
    END
EOF1
set -v
#
# File './CARDIN' contains card-image input for COMMIX.
# See Input Guide in Appendix A.
set +v
cat <<'EOF3' >CARDIN
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
EOF3
set -v
#
# File './CMXOLD' is the restart file written by a previous job.
# For IFRES<2: remove the copy command.
# For IFRES>1: supply an input file name for the copy command.
cp $HOME/priorjob.r CMXOLD
#
# #####
# Prepare and execute the code.
# -----
# Preparation involves compiling the routine(s) in file './SOURCE'
# and linking this with the rest of the base code and one set of
# detailed fluid property routines from the object libraries to form
# an executable code version. Compilation listing and load map are
# written to file '$HOME/thisjob.1'. Compilation or linking errors
# will cause the job to abort.
# Execution involves reading card-image input file './CARDIN' and
# optional binary restart file './CMXOLD', performing the requested
# calculation, adding printable output to file '$HOME/thisjob.1',
# and writing optional binary restart and plot files './CMXNEW'
# and './PLTDAT', respectively.
# #####
cat SOURCE >fort.f
cft77 -e isz -d p -m 1 -- fort.f
ERRCFT=$?
cat fort.1 >$HOME/thisjob.1
rm fort.f fort.1
if test $ERRCFT -ne 0; then banner 'ERR-cft'; rm fort.o; exit; fi
segldr -D 'USX=WARNING' -E \
    -l $CMXPROP,$ANLUTIL -M LDROUT,a \
    -o COMMIX.a fort.o $CMXBASE >LDROUT 2>LDRERR
ERRLDR=$?
cat LDRERR
cat LDROUT >>$HOME/thisjob.1
rm fort.o LDRERR LDROUT
if test $ERRLDR -ne 0; then banner 'ERR-ldr'; rm COMMIX.a; exit; fi
./COMMIX.a >>$HOME/thisjob.1
ERRCMX=$?
rm COMMIX.a
if test $ERRCMX -ne 0; then banner 'ERR-cmx'; fi
```

Table C.5 Cray Batch Job to Execute COMMIX (Cont'd)

```

#
# #####
# Epilogue: File Dispositions
# #####
#
# File './CMXNEW' is the binary restart file written by this job;
# only created if IFRES=1, 3, or 4. If this file exists, copy it
# to a non-scratch directory and change the name.
if test -s CMXNEW; then cp CMXNEW $HOME/thisjob.r; rm CMXNEW; fi
#
# File './PLTDAT' is the binary plot file written by this job; only
# created if nonzero entry(s) in array NTPLOT. If this file exists,
# copy it to a non-scratch directory and change the name.
if test -s PLTDAT; then cp PLTDAT $HOME/thisjob.p; rm PLTDAT; fi
#
# Add file '$HOME/thisjob.1' (compilation listing, load map, and
# results of COMMIX execution having FORTRAN-type carriage control
# information in column 1) to 'stdout'. (OPTIONAL)
echo '1' >fort.f; cat $HOME/thisjob.1 >>fort.f; echo '1' >>fort.f
set +vxS; asa fort.f; set -vxS
rm fort.f
#
# Remove temporary files.
rm SOURCE CARDIN
test -s CMXOLD && rm CMXOLD
#

```

Table C.6 Cray Batch Job to Execute Graphics Postprocessor
Codes VECT or TEMP

```
# user=
#
# QSUB -r thisjob -lT 60 -lM 900Kw -eo
#      jobname      time      memory      stderr
#
# Activate full command echo with timestamp.
set -vxs
#
# #####
# COMMIX: Execute Graphics Postprocessor xxxx (VECT or TEMP).
# -----
# Before distributing this jobstream to users, installer should:
# - In 'Prologue': Define shell variable CMXPLOT.
# - In 'segldr' call: Replace use of shell variable DISLIB11 with
#   name of file containing object code for DISSPLA (and anything
#   else special needed for your system).
# - In 'Epilogue': Replace use of 'hardcopy ...' with proper
#   commands needed to send './popfil.dat' to a real device.
# Before submitting this job, user should:
# - Change 'thisjob' to an actual jobname.
# - In 'Prologue' section: insure shell variables 'TMPDIR' and
#   'CMXPLOT' are properly set and used; properly define files
#   './SOURCE', './CARDIN', and './fort.9'.
# - In 'Epilogue' section: adjust file dispositions.
# -----
# Assumes running under UNICOS (Bourne shell) as NQS batch job.
# Permanent COMMIX system object files read by this job:
# $CMXPLOT: graphics postprocessor object library.
# Shell variable 'CMXPLOT' pointing to COMMIX file is defined in
# 'Prologue' section below.
# Permanent user files used by this job:
# $HOME/cmxjob.*: binary restart (*=r) or plot (*=p) file written
#                 by a previous COMMIX execution job (input).
# $HOME/thisjob.l: printable output with carriage control.
# Temporary files created and deleted by this job are all in the
# directory set in 'Prologue' section using shell variable 'TMPDIR':
# CARDIN, fort.a, fort.f, fort.l, fort.o, fort.9, LDRERR, LDROUT,
# popfil.dat, SOURCE.
# #####
# #####
# Prologue: Directory and File Definitions
# #####
#
# Set current working directory for this job to hold all temporary
# files. Assumes that shell variable 'TMPDIR' exists and is the
# complete path name of this directory, the directory exists, and
# you have full access to it.
cd $TMPDIR
#
# Define shell variable 'CMXPLOT' pointing to object library for
# graphics postprocessor 'xxxx'; definition depends on how code was
# installed on your system; change 'xxxx' to 'VECT' or 'TEMP'.
CMXPLOT='/owner_id_info/COMMIXIARP/OBJECT/xxxx'
#
# File './SOURCE' contains routines to be compiled. Subroutine
# ALTER must always be present; adjust dimension of array S for size
# of problem. Define shell variables containing job name, job
# number, and user name for use in ALTER.
TJNAME='echo "$QSUB_REQNAME      " | cut -c1-8'
```

Table C.6 Cray Batch Job to Execute Graphics Postprocessor
Codes VECT or TEMP (Cont'd)

```

TJNUM='echo "$QSUB REQID      " | cut -c1-8'
TROUTE='echo "$LOGNAME      " | cut -c1-8'
set +v
cat <<EOF1 >SOURCE
    SUBROUTINE ALTER(NAVAIL)
    COMMON /SPACE/ S(100000),SPAEND
    CHARACTER*8 LJNAME,LJNUM,LROUTE
    COMMON /JINFO/ LJNAME,LJNUM,LROUTE
    NAVAIL=(LOC (SPAEND)-LOC (S(1)))/1
    LJNAME='$TJNAME'
    LJNUM='$TJNUM'
    LROUTE='$TROUTE'
    RETURN
    END
EOF1
set -v
#
# File './CARDIN' contains card-image input for postprocessor.
# See Input Guide in Appendix D.
set +v
cat <<'EOF3' >CARDIN
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
EOF3
set -v
#
# File './fort.9' is the binary restart (ITAPE=0) or plot (ITAPE>0)
# written by a previous COMMIX job; supply input file name for copy.
cp $HOME/cmxjob.r fort.9
#
# #####
# Prepare and execute the code.
# -----
# Preparation involves compiling the routine(s) in file './SOURCE'
# and linking this with the rest of the postprocessor code and
# system routines to form an executable code version. Compilation
# listing and load map are written to file '$HOME/thisjob.1'.
# Compilation or linking errors will cause the job to abort.
# Execution involves reading card-image input file './CARDIN' and
# binary restart or plot file './fort.9', creating the requested
# figures, adding printable output to file '$HOME/thisjob.1',
# and writing graphics output in DISSPOP MetaFile format on file
# './popfil.dat'.
# #####
cat SOURCE >fort.f
cft77 -e isz -d p -m 1 -- fort.f
ERRCFT=$?
cat fort.1 >$HOME/thisjob.1
rm fort.f fort.1
if test $ERRCFT -ne 0; then banner 'ERR-cft'; rm fort.o; exit; fi
segldr -D 'USX=WARNING' -E \
    -l $DISLIB11 -M LDROUT,a \
    -o fort.a fort.o $CMXPLOT >LDROUT 2>LDRERR
ERRLDR=$?
cat LDRERR
cat LDROUT >>$HOME/thisjob.1
rm fort.o LDRERR LDROUT
if test $ERRLDR -ne 0; then banner 'ERR-ldr'; rm fort.a; exit; fi
./fort.a <CARDIN >>$HOME/thisjob.1
ERRGO=$?

```

Table C.6 Cray Batch Job to Execute Graphics Post processor
Codes VECT or TEMP (Cont'd)

```

rm fort.a fort.9
if test $ERRGO -ne 0; then banner 'ERR-go'; fi
#
# #####
# Epilogue: File Dispositions
# #####
#
# File './popfil.dat' is the DISSPOP MetaFile format graphics file
# created by this job; if this file exists, send it to a real
# device. At ANL, this is done using the locally written system
# script "hardcopy".
if test -s popfil.dat; then hardcopy popfil.dat ps; rm popfil.dat; fi
#
# Add file '$HOME/thisjob.1' (compilation listing, load map, and
# results of VECT or TEMP execution having FORTRAN-type carriage
# control information in column 1) to 'stdout'. (OPTIONAL)
echo '1' >fort.f; cat $HOME/thisjob.1 >>fort.f; echo '1' >>fort.f
set +vxS; asa fort.f; set -vxS
rm fort.f
#
# Remove temporary files.
rm SOURCE CARDIN
#

```

Table C.7 Cray Batch Job to Execute Graphics Postprocessor Code TDSC

```
# user=
#
# QSUB -r thisjob -lT 60 -lM 900Kw -eo
#       jobname      time      memory      stderr
#
# Activate full command echo with timestamp.
set -vXS
#
# #####
# COMMIX: Execute Graphics Postprocessor TDSC.
# -----
# Before distributing this jobstream to users, installer should:
# - In 'segldr' call: Specify the name of the file containing
#   the object library for TDSC.
# Before submitting this job, user should:
# - Change 'thisjob' to an actual jobname.
# - In 'Prologue' section: insure shell variable 'TMPDIR' is
#   properly used; properly define files './SOURCE', './CARDIN1',
#   './CARDIN2', and './fort.21' through './fort.99'.
# - In 'Epilogue' section: adjust file dispositions.
# -----
# Assumes running under UNICOS (Bourne shell) as NQS batch job.
# Permanent COMMIX system object files read by this job:
# owner_id_info/COMMIX1ARP/OBJECT/TDSC .
# Permanent user files used by this job:
# $HOME/cmxjob*.p: binary plot file(s) written by previous COMMIX
#                   job(s) (input).
# $HOME/thisjob.1: printable output with carriage control.
# $HOME/thisjob.e: binary file created by extraction portion of
#                   this job in format suitable for input to GRPH.
# Temporary files created and deleted by this job are all in the
# directory set in 'Prologue' section using shell variable 'TMPDIR':
# CARDIN1, CARDIN2, fort.a, fort.f, fort.l, fort.o, fort.16,
# fort.21 through fort.99, LDRERR, LDROUT, SOURCE.
# #####
# #####
# Prologue: Directory and File Definitions
# #####
#
# Set current working directory for this job to hold all temporary
# files. Assumes that shell variable 'TMPDIR' exists and is the
# complete path name of this directory, the directory exists, and
# you have full access to it.
cd $TMPDIR
#
# File './SOURCE' contains routines to be compiled. Subroutine
# ALTER must always be present; adjust dimension of array S for size
# of problem.
set +v
cat <<'EOF1' >SOURCE
    SUBROUTINE ALTER(NAVAIL)
    COMMON /SPACE/ S(100000),SPAEND
    NAVAIL=(LOC (SPAEND)-LOC (S(1)))/1
    RETURN
    END
EOF1
set -v
#
# File './CARDIN1' contains card-image input for concatenation mode
# (ITAPE>0) of postprocessor. See Input Guide in Appendix D.
```

Table C.7 Cray Batch Job to Execute Graphics Postprocessor Code TDSC (Cont'd)

```

set +v
cat <<'EOF3' >CARDIN1
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
EOF3
set -v
#
# File './CARDIN2' contains card-image input for extraction mode
# (ITAPE=0) of postprocessor. See Input Guide in Appendix D.
set +v
cat <<'EOF4' >CARDIN2
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
EOF4
set -v
#
# Files './fort.nn' (nn=21,22,...,99) are the input binary plot
# file(s), written by previous COMMIX execution job(s), to be
# used in the concatenation part of this job. Include ITAPE number
# of copy commands, one for each file './fort.nn' in the sequence
# from nn=21 through nn=20+ITAPE. The data on the file indicated by
# 'nn+1' must be the part of the case history which follows the
# data on the file indicated by 'nn'. Maximum of 79 files.
cp $HOME/cmxjob01.p fort.21
cp $HOME/cmxjob02.p fort.22
cp .....
cp $HOME/cmxjobmm.p fort.nn
cp .....
cp $HOME/cmxjob79.p fort.99
#
# #####
# Prepare and execute the code.
# -----
# Preparation involves compiling the routine(s) in file './SOURCE'
# and linking this with the rest of the postprocessor code and
# system routines to form an executable code version. Compilation
# listing and load map are written to file '$HOME/thisjob.1'.
# Compilation or linking errors will cause the job to abort.
# There are two execution steps: concatenation and extraction.
# Concatenation involves reading card-image input file './CARDIN1'
# and binary plot files './fort.21' through './fort.99', creating
# the concatenation of these binary files on file './fort.16'.
# Extraction involves reading card-image input file './CARDIN2' and
# the binary file created by the concatenation step (renamed to be
# './fort.21'), creating binary extraction file './fort.16'. Both
# steps' printable output is appended to file '$HOME/thisjob.1'.
# #####
cat SOURCE >fort.f
cft77 -e isz -d p -m 1 -- fort.f
ERRCFT=$?
cat fort.1 >$HOME/thisjob.1
rm fort.f fort.1
if test $ERRCFT -ne 0; then banner 'ERR-cft'; rm fort.o; exit; fi
segldr -D 'USX=WARNING' -E \
-M LDROUT,a -o fort.a fort.o \
owner_id_info/COMMIX1ARP/OBJECT/TDSC >LDROUT 2>LDRERR
ERRLDR=$?
cat LDRERR
cat LDROUT >>$HOME/thisjob.1
rm fort.o LDRERR LDROUT

```

Table C.7 Cray Batch Job to Execute Graphics Postprocessor Code TDSC (Cont'd)

```

if test $ERRLDR -ne 0; then banner 'ERR-ldr'; rm fort.a; exit; fi
./fort.a <CARDIN1 >>$HOME/thisjob.1 # Concatenate #
ERRGO=$?
rm fort.[2-9][0-9]
if test $ERRGO -ne 0; then banner 'ERR-gol'; rm fort.a; exit; fi
mv fort.16 fort.21
./fort.a <CARDIN2 >>$HOME/thisjob.1 # Extract #
ERRGO=$?
rm fort.a
if test $ERRGO -ne 0; then banner 'ERR-go2'; fi
#
# #####
# Epilogue: File Dispositions
# #####
#
# File './fort.16' is the binary plot file created by the
# extraction step of this job; if this file exists, copy it to a
# non-scratch directory and change the name.
if test -s fort.16; then cp fort.16 $HOME/thisjob.e; rm fort.16; fi
#
# File './fort.21' is the concatenation of the binary plot files
# originally identified as './fort.21' through './fort.99'; if you
# want to save this file, uncomment the next line. (OPTIONAL)
# cp fort.21 $HOME/thisjob.c; rm fort.21
#
# Add file '$HOME/thisjob.1' (compilation listing, load map, and
# results of TDSC execution having FORTRAN-type carriage control
# information in column 1) to 'stdout'. (OPTIONAL)
echo '1' >fort.f; cat $HOME/thisjob.1 >>fort.f; echo '1' >>fort.f
set +vxS; asa fort.f; set -vxS
rm fort.f
#
# Remove temporary files.
rm SOURCE CARDIN1 CARDIN2 fort.21
#

```

Table C.8 Cray Batch Job to Execute Graphics Postprocessor Code GRPH

```

# user=
#
# QSUB -r thisjob -lT 60 -lM 900Kw -eo
#      jobname      time      memory      stderr
#
# Activate full command echo with timestamp.
set -vxS
#
# #####
# COMMIX: Execute Graphics Postprocessor GRPH.
# -----
# Before distributing this jobstream to users, installer should:
# - In 'segldr' call: Replace use of shell variable DISLIB11 with
#   name of file containing object code for DISSPLA (and anything
#   else special needed for your system). Specify the name of
#   the file containing the object library for GRPH.
# - In 'Epilogue': Replace use of 'hardcopy ...' with proper
#   commands needed to send './popfil.dat' to a real device.
# Before submitting this job, user should:
# - Change 'thisjob' to an actual jobname.
# - In 'Prologue' section: insure shell variable 'TMPDIR' is
#   properly used; properly define files './SOURCE', './CARDIN',
#   and './fort.nn' for nn=12, 13, 14, 15, and 16.
# - In 'Epilogue' section: adjust file dispositions.
# -----
# Assumes running under UNICOS (Bourne shell) as NQS batch job.
# Permanent COMMIX system object files read by this job:
# owner_id_info/COMMIX1ARP/OBJECT/GRPH .
# Permanent user files used by this job:
# $HOME/tdscjob*.e: binary file(s) written by previous TDSC
#   extraction job(s) (input).
# $HOME/thisjob.1: printable output with carriage control.
# Temporary files created and deleted by this job are all in the
# directory set in 'Prologue' section using shell variable 'TMPDIR':
# CARDIN, fort.a, fort.f, fort.l, fort.o, fort.12, fort.13,
# fort.14, fort.15, fort.16, LDRERR, LDROUT, popfil.dat, SOURCE.
# #####
# #####
# Prologue: Directory and File Definitions
# #####
#
# Set current working directory for this job to hold all temporary
# files. Assumes that shell variable 'TMPDIR' exists and is the
# complete path name of this directory, the directory exists, and
# you have full access to it.
cd $TMPDIR
#
# File './SOURCE' contains routines to be compiled. Subroutine
# ALTER must always be present; adjust dimension of array S for size
# of problem. Define shell variables containing job name, job
# number, and user name for use in ALTER.
TJNAME='echo "$QSUB REQNAME" | cut -c1-8'
TJNUM='echo "$QSUB REQID" | cut -c1-8'
TROUTE='echo "$LOGNAME" | cut -c1-8'
set +v
cat <<EOF1 >SOURCE
SUBROUTINE ALTER(NAVAIL)
COMMON /SPACE/ S(100000),SPAEND
CHARACTER*8 LJNAME,LJNUM,LROUTE
COMMON /JINFO/ LJNAME,LJNUM,LROUTE

```

Table C.8 Cray Batch Job to Execute Graphics Postprocessor Code GRPH (Cont'd)

```

NAvail=(LOC (SPAEND)-LOC (S(1)))/1
LJNAME=' $TJNAME'
LJNUM=' $TJNUM'
LROUTE=' $TROUTE'
RETURN
END
EOF1
set -v
#
# File './CARDIN' contains card-image input for postprocessor.
# See Input Guide in Appendix D.
set +v
cat <<'EOF3' >CARDIN
. . . . . Replace with
. . . . . actual card-image
. . . . . input file
EOF3
set -v
#
# Files './fort.nn' are the primary (nn=15) and first alternate (nn=16),
# second alternate (nn=12), third alternate (nn=13), and fourth
# alternate (nn=14) binary input files as prepared by previous TDSC
# extraction runs; at least one of these files must be defined.
# If IFFT15=0: remove the fort.15 copy command.
# If IFFT15=1: supply input file name for the fort.15 copy command.
# If IFFT16=0: remove the fort.nn copy commands for nn=16,12,13,14.
# If IFFT16>0: supply input file name for the first IFFT16 of the
# fort.nn copy commands for nn=16,12,13,14 and remove the others.
cp $HOME/tdscjob1.e fort.15
cp $HOME/tdscjob2.e fort.16
cp $HOME/tdscjob3.e fort.12
cp $HOME/tdscjob4.e fort.13
cp $HOME/tdscjob5.e fort.14
#
# #####
# Prepare and execute the code.
# -----
# Preparation involves compiling the routine(s) in file './SOURCE'
# and linking this with the rest of the postprocessor code and
# system routines to form an executable code version. Compilation
# listing and load map are written to file '$HOME/thisjob.1'.
# Compilation or linking errors will cause the job to abort.
# Execution involves reading card-image input file './CARDIN' and
# binary files './fort.15', './fort.16', './fort.12', './fort.13',
# and/or './fort.14', creating the requested figures, adding printable
# output to file '$HOME/thisjob.1', and writing graphics output in
# DISSPOP MetaFile format on file 'popfil.dat'.
# #####
cat SOURCE >fort.f
cft77 -e isz -d p -m 1 -- fort.f
ERRCFT=$?
cat fort.1 >$HOME/thisjob.1
rm fort.f fort.1
if test $ERRCFT -ne 0; then banner 'ERR-cft'; rm fort.o; exit; fi
segldr -D 'USX=WARNING' -E -l $DISLIB11 \
-M LDROUT,a -o fort.a fort.o \
owner id_info/COMMIX1ARP/OBJECT/GRPH >LDROUT 2>LDRERR
ERRLDR=$?
cat LDRERR
cat LDROUT >>$HOME/thisjob.1
rm fort.o LDRERR LDROUT
if test $ERRLDR -ne 0; then banner 'ERR-ldr'; rm fort.a; exit; fi
./fort.a <CARDIN >>$HOME/thisjob.1
ERRGO=$?
rm fort.a fort.15 fort.16 fort.12 fort.13 fort.14
if test $ERRGO -ne 0; then banner 'ERR-go'; fi
#
# #####
# Epilogue: File Dispositions

```

Table C.8 Cray Batch Job to Execute Graphics Postprocessor Code GRPH (Cont'd)

```

# #####
#
# File './popfil.dat' is the DISSPOP MetaFile format graphics file
# created by this job; if this file exists, send it to a real
# device. At ANL, this is done using the locally written system
# script "hardcopy".
if test -s popfil.dat; then hardcopy popfil.dat ps; rm popfil.dat; fi
#
# Add file '$HOME/thisjob.1' (compilation listing, load map, and
# results of GRPH execution having FORTRAN-type carriage control
# information in column 1) to 'stdout'. (OPTIONAL)
echo '1' >fort.f; cat $HOME/thisjob.1 >>fort.f; echo '1' >>fort.f
set +vxS; asa fort.f; set -vxS
rm fort.f
#
# Remove temporary files.
rm SOURCE CARDIN
#

```

C.1.2.2 IBM

As a prefacing note, compilation, link editing, and execution are performed by calling various cataloged (system) procedures. The cataloged procedures in use at ANL differ substantially from those supplied by IBM in their use of symbolic parameters, DDNAMEs, and libraries. For this reason, the ANL versions of these cataloged procedures will be shown as part of this installation section. For reference in what follows, the step names used in these procedures are FORTV for compilation, EDT for link editing, and GO for execution. You will need to modify the sample jobstreams to account for these differences between the ANL procedures and those in use at your installation.

A sample jobstream to compile and create an object library on the IBM is shown in Table C.9. This job assumes that the source code for code section "xxxx" is in a cataloged dataset named owner_id_info.CMX1ARP.SOURCE.xxxx, where "owner_id_info" is a valid base-level index for you and "xxxx" is one of the code section names shown in Table C.3. This job will leave the object library on a cataloged dataset named owner_id_info.CMX1ARP.OBJECT.xxxx. Compilation is done using the VS FORTRAN compiler. The compiler is invoked by calling the cataloged procedure named FORTVCP; a listing of this procedure, as implemented at ANL, is shown in Table C.10. The source code to be compiled is identified by the DDNAME of FORTV.SYSIN and the object code created is identified by the DDNAME of FORTV.SYSOBJ. The example shown in Table C.9 assumes that the object code is to be written to a disk file; the SPACE allocation needed for these object files depends on the code section being compiled; based on IBM-3380-type disks, the base COMMIX code needs 75 tracks, each set of detailed fluid properties needs 1 to 2 tracks, each graphics postprocessor code needs 5 to 10 tracks. This job should be submitted for each code section to be installed (changing "xxxx" to the name of the code section).

Sample jobstreams for executing COMMIX and the graphics postprocessors are shown in Tables C.11 through C.14. You will need to modify these before distributing them to the other users at your site. A number of these changes are common to all these samples. Each job contains some combination of compilation, link editing, and execution, which are accomplished

by calling one or more of the cataloged procedures FORTVCEG, FORTVCEP, and RUN. The listings of these three procedures, as implemented at ANL, are given in Tables C.15 through C.17. The primary modifications needed are those connected with the libraries to be used in the link edit step; the various special system routines needed by each code section, along with the names of the datasets which contain these routines at ANL, are identified in Table C.1; the sample jobstreams assume that these special system routines are available as part of the automatic-call linkage editor library referenced by the DDNAME EDT.SYSLIB; you will need to modify the list of datasets being concatenated to form EDT.SYSLIB in the cataloged procedures to reference the libraries containing these routines (or their replacements) on your system. The binary restart and plot files created by COMMIX are assumed to be left on the IBM disks as cataloged datasets named `user_id_info.thisjob.RESTRTI` and `user_id_info.thisjob.PLOT76I`, respectively (where the user is assumed to have replaced "user_id_info" with a valid base-level index and changed the string "thisjob" to the actual job name prior to submittal); you may want to recommend alternate permanent file names and storage locations for your users. You will need to adjust the JOB card parameters (e.g., accounting field) and possibly add `//*MAIN` and `//*FORMAT` cards for your system. The additional changes needed will be discussed for each type of job.

The sample jobstream for executing COMMIX is shown in Table C.11. This compile, link, and execute job invokes the cataloged procedure FORTVCEG shown in Table C.15. You need to make three changes in the set of DDNAMEs used by the linkage editor. (1) The DDNAME EDT.BASE refers to the object library containing the base COMMIX code; you should specify the complete DSN field on this DD card. (2) The DDNAME EDT.PROP refers to the object library containing the one set of detailed fluid properties to be used for a case; you should adjust the initial part of the DSN field on this DD card to conform to how you installed the object code sections and modify the comment in this section to indicate the available choices to the user if all four sets of detailed fluid properties were not installed, leaving it to the user to select one set of properties by completing the file name when running a case. (3) You must modify the definition of DDNAME EDT.SYSLIB to include the libraries which contain LOCF, TLEFT, and LINPACK (or their replacements), as discussed previously in the paragraph on modifications common to all execution samples. You may want to alter the recommended fields on the DD

cards for execution step binary files referenced by DDNAMEs GO.FT09F001, GO.FT10F001, and GO.FT76F001; the SPACE allocation for FT10F001 and FT76F001 depends on the case being run.

The sample jobstream for executing graphics postprocessor codes VECT and TEMP is shown in Table C.12. This compile, link, and execute job invokes the cataloged procedure FORTVCEG shown in Table C.15. You need to make two changes in the set of DDNAMEs used by the linkage editor. (1) The DDNAME EDT.OBJECT refers to the object library containing the code for postprocessor "xxxx"; you should specify the initial part of the DSN field, leaving it for the user to specify either "VECT" or "TEMP" for "xxxx" to select which postprocessor code to run. (2) You must modify the definition of DDNAME EDT.SYSLIB to include the libraries which contain LOCF, DATE1, TIME1, ANLACT, and DISSPLA (or their replacements), as discussed previously in the paragraph on modifications common to all execution samples; note that DISSPLA is referenced via the PRELIB parameter in the call of the FORTVCEG procedure. In connection with the use of DISSPLA, you must also specify the DSN field for DDNAMEs GO.FT44F001 and GO.CADYNLIB to reference the appropriate datasets for your system. At the end of the jobstream, change the use of the "PPS" cataloged procedure to the appropriate program call(s) needed to send the graphics to a real device at your installation. You may want to alter the recommended fields on the DD cards for execution step files referenced by DDNAMEs GO.FT09F001 and GO.FT19F001; the SPACE allocation for FT19F001 depends on the number and complexity of the figures being drawn.

The sample jobstream for executing graphics postprocessor code TDSC is shown in Table C.13. This compile, link, and double execute job invokes the cataloged procedures FORTVCEP and RUN shown in Tables C.16 and C.17, respectively. You need to make two changes in the set of DDNAMEs used by the linkage editor. (1) The DDNAME EDT.OBJECT refers to the object library containing the code for postprocessor TDSC; you should specify the complete DSN field on this DD card. (2) You must modify the definition of DDNAME EDT.SYSLIB to include the library which contains LOCF (or its replacement), as discussed previously in the paragraph on modifications common to all execution samples. (Postprocessor TDSC does not produce any graphics and, thus, does not need DISSPLA; the program is used to prepare the

binary input file read by postprocessor GRPH, which does produce graphical output.) You may want to alter the recommended fields on the DD cards for concatenation and extraction execution step files referenced by DDNAMEs GO.FT09F001 and GO.FT16F001; the SPACE allocation for FT16F001 depends on the amount of data being concatenated and extracted.

The sample jobstream for executing graphics postprocessor code GRPH is shown in Table C.14. This compile, link, and execute job invokes the cataloged procedure FORTVCEG shown in Table C.15. You need to make two changes in the set of DDNAMEs used by the linkage editor. (1) The DDNAME EDT.OBJECT refers to the object library containing the code for postprocessor GRPH; you should specify the complete DSN field on this DD card. (2) You must modify the definition of DDNAME EDT.SYSLIB to include the libraries which contain LOCF and DISSPLA (or their replacements), as discussed previously in the paragraph on modifications common to all execution samples; note that DISSPLA is referenced via the PRELIB parameter in the call of the FORTVCEG procedure. In connection with the use of DISSPLA, you must also specify the DSN field for DDNAMEs GO.FT44F001 and GO.CADYNLIB to reference the appropriate datasets for your system. At the end of the jobstream, change the use of the "PPS" cataloged procedure to the appropriate program call(s) needed to send the graphics to a real device at your installation. You may want to alter the recommended fields on the DD cards for execution step files referenced by DDNAMEs GO.FT12F001, GO.FT13F001, GO.FT14F001 GO.FT15F001, GO.FT16F001, and GO.FT19F001; the SPACE allocation for FT19F001 depends on the number and complexity of the figures being drawn.

Table C.9 IBM Batch Job to Compile Code Section

```
//COMPILE JOB (account_info),TIME=5
/**
/** *****
/** COMMIX: Compile code module 'xxxx' to Object Library.
/** -----
/** Before submitting this job:
/** - Change 'xxxx' to name of appropriate code module.
/** - Replace 'owner_id_info' with valid base-level dsn index.
/** - Adjust JOB and /*MAIN card parameters for your system.
/** -----
/** Assumes running under MVS/JES3.
/** Permanent files used by this job:
/** owner_id_info.CMX1ARP.SOURCE.xxxx: source code (input).
/** owner_id_info.CMX1ARP.OBJECT.xxxx: object code (output).
/** *****
/**
/** *MAIN LINES=50
/**
/** *****
//COMPILE EXEC FORTVCP,OPTIONS='TERM,SOURCE,NOMAP,NOXREF',
// REGN=2400K
/** -----
/** Compile program and save object code.
/** *****
/**
/** *SYSIN: source code for module 'xxxx' (input).
//FORTV.SYSIN DD DSN=owner_id_info.CMX1ARP.SOURCE.xxxx,DISP=SHR
/**
/** *SYSOBJ: object code for module 'xxxx' (output).
//FORTV.SYSOBJ DD DSN=owner_id_info.CMX1ARP.OBJECT.xxxx,
// DISP=(NEW,CATLG),UNIT=DISK,SPACE=(TRK,(80,5),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
```

Table C.10 IBM Cataloged Procedure FORTVCP at ANL for Compilation

```
//FORTVCP PROC COMPILE=FORTVS2,
// INCLUDE='SYS1.DUMMYLIB',
// LVL=77,
// OPTIONS=,
// REGN=2048K,
// STEPLIB='SYS1.VSF2COMP',
// SYSTEM='*'
//FORTV EXEC PGM=&COMPILE,REGION=&REGN,
// PARM='LANGLVL(&LVL),NOTF,&OPTIONS'
//STEPLIB DD DISP=SHR,DSN=&STEPLIB
//SYSLIB DD DISP=SHR,DSN=&INCLUDE
//SYSLIN DD DDNAME=SYSOBJ
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//SYSPUNCH DD SYSOUT=B,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//SYSTEM DD SYSOUT=&SYSTEM,DCB=*.SYSPRINT
```

Table C.11 IBM Batch Job to Execute COMMIX

```

//thisjob JOB (account_info),TIME=60
//*
//* *****
//* COMMIX: Execution.
//* -----
//* Before distributing this jobstream to users, installer should:
//*   - Define DSNs for EDT.BASE and EDT.PROP.
//*   - Adjust usage of cataloged procedure FORTVCEG to conform to
//*     your system, including modifications needed to access LOCF,
//*     TLEFT, and LINPACK (or their replacements).
//*   - Adjust recommended fields on NJB card, EXEC card, and DD
//*     cards for GO.FT10F001 and GO.FT76F001.
//* Before submitting this job, user should:
//*   - Adjust JOB card fields.
//*   - Complete DSN field for EDT.PROP; the allowed names depend
//*     on how COMMIX was installed on your system.
//*   - Add proper in-stream input for FORTV.SYSIN and GO.SYSIN.
//*   - Specify execution memory (GOREGN on EXEC card); estimate:
//*     1800 + 4*(dimension of S)/1000 in Kilo-bytes.
//*   - Adjust DD cards for GO.FT09F001, GO.FT10F001, GO.FT76F001.
//* -----
//* Assumes running under MVS/JES3; submit as "unnumbered".
//* Permanent COMMIX system object files read by this job:
//* owner_id_info.CMX1ARP.OBJECT.BASE: base COMMIX code.
//* owner_id_info.CMX1ARP.OBJECT.PROPxxxx: fluid 'xxxx' properties.
//* Permanent user files used by this job:
//* user_id_info.priorjob.RESTRTI: restart file from prior job (in).
//* user_id_info.thisjob.RESTRTI: restart file created by this job.
//* user_id_info.thisjob.PLOT76I: plot file created by this job.
//* *****
//* -----
//RUN EXEC FORTVCEG,OPTIONS='TERM,SOURCE,NOMAP,NOXREF',
// GOREGN=2000K
//* -----
//* FORTV: Compile routine(s) (at least ALTER): source in SYSIN.
//* EDT: Link with base code, properties, and system routines.
//* GO: Execute the above code:
//* Card input data is on SYSIN and input Restart file is FT09F001;
//* printed output to FT06F001, output Restart file on FT10F001,
//* and output Plot file on FT76F001.
//* *****
//*
//* SYSIN: routines to be compiled. Subroutine ALTER must always
//* be present; adjust dimension of array S for size of problem.
//* Add functions HEATCU, CPSOL1, and XKSOL1 if needed.
//FORTV.SYSIN DD *
      SUBROUTINE ALTER(NAVAIL)
      INTEGER S,SPAEND
      COMMON /SPACE/ S(221000),SPAEND
      COMMON /MACHIN/ NOBPI,NOWPD,NOWPS
      NAVAIL=(LOCF(SPAEND)-LOCF(S(1)))/NOBPI
      RETURN
      END
//
//*
//* BASE: base object code (no detailed fluid properties).
//EDT.BASE DD DSN=owner_id_info.CMX1ARP.OBJECT.BASE,DISP=SHR
//*
//* PROP: detailed properties object code for fluid number 1;
//* change 'xxxx' to 'NA' for sodium, 'H2O' for normal water,
//* 'D2O' for heavy water, or 'HE' for helium.

```

Table C.11 IBM Batch Job to Execute COMMIX (Cont'd)

```
//EDT.PROP DD DSN=owner_id_info.CMX1ARP.OBJECT.PROPxxxx,DISP=SHR
//*
//* SYSIN: directives to linkage editor.
//EDT.SYSIN DD *
REPLACE SPACE
INCLUDE BASE
INCLUDE PROP
ENTRY MAIN
/*
//* SYSIN: card-image input for COMMIX.
//* See Input Guide in Appendix A.
//GO.SYSIN DD *
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
/*
//* FT09F001: binary restart file written by a previous job.
//* For IFRES<2: remove the DD card.
//* For IFRES>1: supply DSN field.
//GO.FT09F001 DD DSN=user_id_info.priorjob.RESTRTI,DISP=SHR
/*
//* FT10F001: binary restart file written by this job.
//* For IFRES=0 or 2: remove DD card.
//* For IFRES=1, 3, or 4: supply DSN and other fields.
//GO.FT10F001 DD DSN=user_id_info.thisjob.RESTRTI,
// DISP=(NEW,CATLG),UNIT=DISK,SPACE=(TRK,(40,20),RLSE),
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6233)
/*
//* FT76F001: binary plot file written by this job.
//* For NTPLOT()=0: remove DD card.
//* For NTPLOT() not 0: supply DSN and other fields.
//GO.FT76F001 DD DSN=user_id_info.thisjob.PLOT76I,
// DISP=(NEW,CATLG),UNIT=DISK,SPACE=(TRK,(40,20),RLSE),
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6233)
```

Table C.12 IBM Batch Job to Execute Graphics Postprocessor
Codes VECT or TEMP

```
//thisjob JOB (account_info),TIME=2
//*
//* *****
//* COMMIX: Execute Graphics Postprocessor xxxx (VECT or TEMP).
//* -----
//* Before distributing this jobstream to users, installer should:
//* - Define DSN for EDT.OBJECT.
//* - Adjust usage of cataloged procedure FORTVCEG to conform to
//*   your system, including modifications needed to access LOCF,
//*   DATE1, TIME1, ANLACT, and DISPSLA (or their replacements).
//* - Adjust DSN fields for GO.FT44F001 and GO.CADYNLIB.
//* - Replace " EXEC PPS,..." with appropriate program call.
//* - Adjust recommended fields on NJB card, EXEC card, and DD
//*   card for GO.FT19F001.
//* Before submitting this job, user should:
//* - Adjust JOB card fields.
//* - Change 'xxxx' to 'VECT' or 'TEMP' in DSN field of EDT.OBJECT
//*   and on ENTRY directive in EDT.SYSIN.
//* - Add proper in-stream input for FORTV.SYSIN and GO.SYSIN.
//* - Specify execution memory (GOREGN on EXEC card); estimate:
//*   950 + 4*(dimension of S)/1000 in Kilo-bytes.
//* - Adjust DD cards for GO.FT09F001 and GO.FT19F001.
//* -----
//* Assumes running under MVS/JES3; submit as "unnumbered".
//* Permanent COMMIX system object files read by this job:
//* owner_id_info.CMX1ARP.OBJECT.xxxx: code for 'xxxx'.
//* Permanent user files used by this job:
//* user_id_info.cmxrjob.*: binary restart (*=RESTRTI) or plot
//*   (*=PLOT76I) file written by a previous COMMIX execution job.
//* *****
//* *****
//DRAW EXEC FORTVCEG,OPTIONS='TERM,SOURCE,NOMAP,NOXREF',
//PRELIB='CA.DIS110.DISLIB.LOAD',
//GOREGN=1350K
//* -----
//* FORTV: Compile ALTER: source in SYSIN.
//* EDT: Link with rest of code and system routines.
//* GO: Execute the above code:
//* Card input data is on SYSIN; input binary restart or plot file
//* or plot file from COMMIX is on FT09F001; printed output to
//* FT06F001; output graphics to FT19F001.
//* *****
//*
//* SYSIN: source code for Subroutine ALTER; adjust dimension of
//* array S for size of problem.
//FORTV.SYSIN DD *
        SUBROUTINE ALTER(NAVAIL)
        COMMON /SPACE/ S(100000),SPAEND
        NAVAIL=(LOC(SPAEND)-LOC(S(1)))/4
        RETURN
        END
//*
//* OBJECT: complete object library for postprocessor 'xxxx';
//* change 'xxxx' to 'VECT' or 'TEMP'.
//EDT.OBJECT DD DSN=owner_id_info.CMX1ARP.OBJECT.xxxx,DISP=SHR
//*
//* SYSIN: directives to linkage editor.
//* Change 'xxxx' to either 'VECT' or 'TEMP' on ENTRY directive.
//EDT.SYSIN DD *
```

Table C.12 IBM Batch Job to Execute Graphics Postprocessor
Codes VECT or TEMP (Cont'd)

```

REPLACE SPACE
INCLUDE OBJECT
ENTRY AAxxxx
/*
/** FT09F001: input binary restart (ITAPE=0) or plot (ITAPE>0) file
/** as written by a previous COMMIX execution job; specify DSN.
//GO.FT09F001 DD DSN=user_id_info.cmxjob.RESTRTI,DISP=SHR
/**
/** FT19F001: output graphics file written by DISSPLA in DISSPOP
/** metafile format; assumed to be passed to next job step.
//GO.FT19F001 DD DSN=&META,DISP=(NEW,PASS),
// UNIT=SASCR,SPACE=(TRK,(100,20),RLSE)
/**
/** FT44F001: dynamic device driver information for DISSPLA.
//GO.FT44F001 DD DSN=CA.DIS110.DRVS.GRAFPROD.SITEDATA,DISP=SHR,
// LABEL=(,,,IN)
/**
/** CADYNLIB: dynamic device driver information for DISSPLA.
//GO.CADYNLIB DD DSN=CA.DIS110.DRVS.DYNLIB.LOAD,DISP=SHR
/**
/** SYSIN: card-image input for postprocessor.
/** See Input Guide in Appendix D.
//GO.SYSIN DD *
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
/*
/** *****
//POST EXEC PPS,INDSN='&META',DEST='RM113PR2',REGN=1024K
/** -----
/** Move results on INDSN to a PostScript device at DEST. At ANL,
/** this is accomplished using a locally written procedure "PPS".
/** *****

```

Table C.13 IBM Batch Job to Execute Graphics Postprocessor
Code TDSC

```
//thisjob JOB (account_info),TIME=2
//*
/* *****
/* COMMIX: Execute Graphics Postprocessor TDSC.
/* -----
/* Before distributing this jobstream to users, installer should:
/* - Define DSN for EDT.OBJECT.
/* - Adjust usage of cataloged procedure FORTVCEP to conform to
/* your system, including modifications needed to access LOCF
/* (or its replacement).
/* - Adjust usage of cataloged procedure RUN for your system.
/* - Adjust recommended fields on NJB card, EXEC cards, and DD
/* cards for GO.FT16F001 in both steps which call RUN.
/* Before submitting this job, user should:
/* - Adjust JOB card fields.
/* - Add proper in-stream input for FORTV.SYSIN and both
/* occurrences of GO.SYSIN.
/* - Specify execution memory (GOREGN on EXEC cards); estimate:
/* 150 + 4*(dimension of S)/1000 in Kilo-bytes.
/* - Adjust DD cards for GO.FT09Fnnn in the CONCAT step and
/* GO.FT16F001 in the XTRACT step.
/* -----
/* Assumes running under MVS/JES3; submit as "unnumbered".
/* Permanent COMMIX system object files read by this job:
/* owner_id_info.CMX1ARP.OBJECT.TDSC: code for 'TDSC'.
/* Permanent user files used by this job:
/* user_id_info.cmxjob*.PLOT76I: binary plot file(s) written by
/* previous COMMIX execution job(s).
/* *****
/* -----
/* COMPILE EXEC FORTVCEP,OPTIONS='TERM,SOURCE,NOMAP,NOXREF'
/* -----
/* FORTV: Compile ALTER: source in SYSIN.
/* EDT: Link with rest of code and system routines:
/* Save load module on SYSPVT.
/* *****
/* SYSIN: source code for Subroutine ALTER; adjust dimension of
/* array S for size of problem.
//FORTV.SYSIN DD *
SUBROUTINE ALTER(NAVAIL)
COMMON /SPACE/ S(100000),SPAEND
NAVAIL=(LOC(SPAEND)-LOC(S(1)))/4
RETURN
END
/*
/* OBJECT: complete object library for postprocessor TDSC.
//EDT.OBJECT DD DSN=owner_id_info.CMX1ARP.OBJECT.TDSC,DISP=SHR
/*
/* SYSIN: directives to linkage editor.
//EDT.SYSIN DD *
REPLACE SPACE
INCLUDE OBJECT
ENTRY AATDSC
NAME TDSC
/*
/* SYSPVT: load module created by this step; pass to next steps.
//EDT.SYSPVT DD DSN=&LOAD2,DISP=(NEW,PASS),
// UNIT=SASCR,SPACE=(TRK,(50,20,3),RLSE),DCB=BLKSIZE=6144
```

Table C.13 IBM Batch Job to Execute Graphics Postprocessor
Code TDSC (Cont'd)

```

//*
//* *****
//CONCAT EXEC RUN,M=TDSC,DSN='&LOAD2',GOREGN=550K,DUMP=NULLFILE
//* -----
//* Concatenate 2 (or more) plot files from FT09Fnnn onto FT16F001
//* under control of directives in SYSIN.
//* *****
//*
//* FT09Fnnn: binary input plot files as written by previous COMMIX
//* execution jobs; include ITAPE number of DD cards for nnn=001
//* through nnn=ITAPE. The data on the file referenced by "nnn+1"
//* must be the part of the case history which follows the data on
//* the file referenced by "nnn".
//GO.FT09F001 DD DSN=user_id_info.cmxjob001.PLOT76I,DISP=SHR
//GO.FT09F002 DD DSN=user_id_info.cmxjob002.PLOT76I,DISP=SHR
//GO.FT09F... ..
//GO.FT09Fnnn DD DSN=user_id_info.cmxjobnnn.PLOT76I,DISP=SHR
//GO.FT09F... ..
//GO.FT09Fnnnn DD DSN=user_id_info.cmxjobnnnn.PLOT76I,DISP=SHR
//*
//* FT16F001: binary output file containing the concatenation of
//* the input files from FT09Fnnn; pass to next step.
//GO.FT16F001 DD DSN=&CONCAT,DISP=(NEW,PASS),
// UNIT=SASCR,SPACE=(CYL,(10,5),RLSE),
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6233)
//*
//* SYSIN: card-image input for concatenation mode (ITAPE>0) of
//* postprocessor; see Input Guide in Appendix D.
//GO.SYSIN DD *
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
/*
//* *****
//XTRACT EXEC RUN,M=TDSC,DSN='&LOAD2',GOREGN=550K,DUMP=NULLFILE
//* -----
//* Extract data from concatenation file on FT09F001 and place it
//* on FT16F001 in GRPH format under control of directives on SYSIN.
//* *****
//*
//* FT09F001: binary input file created by CONCAT step.
//GO.FT09F001 DD DSN=&CONCAT,DISP=(OLD,DELETE)
//*
//* FT16F001: binary output file containing the extraction of data
//* from the input file FT09F001; save for use by GRPH.
//GO.FT16F001 DD DSN=user_id_info.thisjob.EXTRACTI,
// DISP=(NEW,CATLG),UNIT=DISK,SPACE=(TRK,(40,20),RLSE),
// DCB=(RECFM=VBS,LRECL=X,BLKSIZE=6233)
//*
//* SYSIN: card-image input for extraction mode (ITAPE=0) mode
//* of postprocessor; see Input Guide in Appendix D.
//GO.SYSIN DD *
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
/*

```

Table C.14 IBM Batch Job to Execute Graphics Postprocessor Code GRPH

```

//thisjob JOB (account_info),TIME=2
//*
//* *****
//* COMMIX: Execute Graphics Postprocessor GRPH.
//* -----
//* Before distributing this jobstream to users, installer should:
//*   - Define DSN for EDT.OBJECT.
//*   - Adjust usage of cataloged procedure FORTVCEG to conform to
//*     your system, including modifications needed to access LOCF,
//*     DATE1, TIME1, ANLACT, and DISPSLA (or their replacements).
//*   - Adjust DSN fields for GO.FT44F001 and GO.CADYNLIB.
//*   - Replace " EXEC PPS,..." with appropriate program call.
//*   - Adjust recommended fields on NJB card, EXEC card, and DD
//*     card for GO.FT19F001.
//* Before submitting this job, user should:
//*   - Adjust JOB card fields.
//*   - Add proper in-stream input for FORTV.SYSIN and GO.SYSIN.
//*   - Specify execution memory (GOREGN on EXEC card); estimate:
//*     200 + 4*(dimension of S)/1000 in Kilo-bytes.
//*   - Adjust DD cards for GO.FT12F001, GO.FT13F001, GO.FT14F001,
//*     GO.FT15F001, GO.FT16F001, and GO.FT19F001.
//* -----
//* Assumes running under MVS/JES3; submit as "unnumbered".
//* Permanent COMMIX system object files read by this job:
//* owner_id_info.CMX1ARP.OBJECT.GRPH: code for 'GRPH'.
//* Permanent user files used by this job:
//* user_id_info.tdscjob*.EXTRACTI: binary file written by previous
//* extraction-mode execution of TDSC.
//* *****
//* *****
//DRAW EXEC FORTVCEG,OPTIONS='TERM,SOURCE,NOMAP,NOXREF',
//  PRELIB='CA.DIS110.DISLIB.LOAD',
//  GOREGN=600K
//* -----
//* FORTV: Compile ALTER: source in SYSIN.
//* EDT: Link with rest of code and system routines.
//* GO: Execute the above code:
//* Card input data is on SYSIN; input binary extraction files from
//* TDSC are on FT15F001, FT16F001, FT12F001, FT13F001, and/or
//* FT14F001; printed output to FT06F001; output graphics to
//* FT19F001.
//* *****
//*
//* SYSIN: source code for Subroutine ALTER; adjust dimension of
//* array S for size of problem.
//FORTV.SYSIN DD *
//  SUBROUTINE ALTER(NAVAIL)
//  COMMON /SPACE/ S(100000),SPAEND
//  NAVAIL=(LOC(SPAEND)-LOC(S(1)))/4
//  RETURN
//  END
//*
//* OBJECT: complete object library for postprocessor GRPH.
//EDT.OBJECT DD DSN=owner_id_info.CMX1ARP.OBJECT.GRPH,DISP=SHR
//*
//* SYSIN: directives to linkage editor.
//EDT.SYSIN DD *
//  REPLACE SPACE
//  INCLUDE OBJECT
//  ENTRY AAGRPH
//*

```

Table C.14 IBM Batch Job to Execute Graphics Postprocessor Code GRPH (Cont'd)

```

/** FTnnF001: primary (nn=15), first alternate (nn=16), second
/** alternate (nn=12), third alternate (nn=13), and fourth alternate
/** (nn=14) binary input files as prepared by previous TDSC extraction
/** runs; a DD card must be present for a least one of these files.
/** If IFFT15=0: remove the DD card for FT15F001.
/** If IFFT15=1: supply the DSN field on DD card for FT15F001.
/** If IFFT16=0: remove the DD card for FTnnF001 for nn=16,12,13,14.
/** If IFFT16>0: supply the DSN field on DD card for the first IFFT16
/** of FTnnF001 for nn=16,12,13,14 and remove the others.
//GO.FT15F001 DD DSN=user_id_info.tdscjob1.EXTRACTI,DISP=SHR
//GO.FT16F001 DD DSN=user_id_info.tdscjob2.EXTRACTI,DISP=SHR
//GO.FT12F001 DD DSN=user_id_info.tdscjob3.EXTRACTI,DISP=SHR
//GO.FT13F001 DD DSN=user_id_info.tdscjob4.EXTRACTI,DISP=SHR
//GO.FT14F001 DD DSN=user_id_info.tdscjob5.EXTRACTI,DISP=SHR
/**
/** FT19F001: output graphics file written by DISSPLA in DISSPOP
/** metafile format; assumed to be passed to next job step.
//GO.FT19F001 DD DSN=&META,DISP=(NEW,PASS),
// UNIT=SASCR,SPACE=(TRK,(100,20),RLSE)
/**
/** FT44F001: dynamic device driver information for DISSPLA.
//GO.FT44F001 DD DSN=CA.DIS110.DRVS.GRAFPDOD.SITEDATA,DISP=SHR,
// LABEL=(,,IN)
/**
/** CADYNLIB: dynamic device driver information for DISSPLA.
//GO.CADYNLIB DD DSN=CA.DIS110.DRVS.DYNLIB.LOAD,DISP=SHR
/**
/** SYSIN: card-image input for postprocessor.
/** See Input Guide in Appendix D.
//GO.SYSIN DD *
. . . . . Replace with . . . . .
. . . . . actual card-image . . . . .
. . . . . input file . . . . .
/**
/** *****
//POST EXEC PPS,INDSN='&META',DEST='RM113PR2',REGN=1024K
/** -----
/** Move results on INDSN to a PostScript device at DEST. At ANL,
/** this is accomplished using a locally written procedure "PPS".
/** *****

```

Table C.15 IBM Cataloged Procedure FORTVCEG at ANL for Compile,
Link Edit, and Execution

```
//FORTVCEG PROC AMDLIB='SYS1.AMDLIB',
//          AMDLIBX='SYS1.AMDLIBX',
//          COMPILE=FORTVS2,
//          EDITOR=IEWL,
//          EDTIF='(5,LT,FORTV)',
//          EDTOPTS=,
//          EDTREGN=512K,
//          E1ROOM='(CYL,(1,4))',
//          GOIF='(5,LT,FORTV),(5,LT,EDT)',
//          GOLIB='SYS1.VSF2LOAD',
//          GOPARM=ABSDUMP,
//          GOREGN=512K,
//          INCLUDE='SYS1.DUMMYLIB',
//          LDDISP='(MOD,PASS)',
//          LDROOM='(CYL,(1,4,1))',
//          LDUNIT=SASCR,
//          LIBRARY='SYS1.VSF2FORT',
//          LOAD='&G(G)',
//          LSIZE='(8192K,64K)',
//          LVL=77,
//          OBJBLK=3120,
//          OBJDISP='(OLD,DELETE)',
//          OBJECT='&OBJT',
//          OBJROOM='(CYL,(10,5))',
//          OBJUNIT=SASCR,
//          OPTIONS=,
//          PLOTUTL='SYS1.PLOTUTL',
//          POSTLIB='SYS1.DUMMYLIB',
//          PRELIB='SYS1.DUMMYLIB',
//          PRELIB2='SYS1.DUMMYLIB',
//          REGN=2048K,
//          STEPLIB='SYS1.VSF2COMP',
//          SYSTEM='*'
//FORTV EXEC PGM=&COMPILE,REGION=&REGN,
//          PARM='LANGLVL(&LVL),NOTF,&OPTIONS'
//STEPLIB DD DISP=SHR,DSN=&STEPLIB
//SYSLIB DD DISP=SHR,DSN=&INCLUDE
//SYSLIN DD UNIT=&OBJUNIT,DSNAME=&OBJECT,SPACE=&OBJROOM,
//          DISP=(MOD,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=&OBJBLK)
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//SYSPUNCH DD SYSOUT=B,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//SYSTEM DD SYSOUT=&SYSTEM,DCB=*.SYSPRINT
//EDT EXEC PGM=&EDITOR,PARM='DCBS,LIST,MAP,&EDTOPTS,SIZE=&LSIZE',
//          REGION=&EDTREGN,COND=&EDTIF
//SYSLIB DD DISP=SHR,DSN=&PRELIB
//          DD DISP=SHR,DSN=&PRELIB2
//          DD DISP=SHR,DSN=&LIBRARY
//          DD DISP=SHR,DSN=&AMDLIB
//          DD DISP=SHR,DSN=&AMDLIBX
//          DD DISP=SHR,DSN=&POSTLIB
//          DD DISP=SHR,DSN=&PLOTUTL
//SYSLIN DD DSN=*.FORTV.SYSLIN,DISP=&OBJDISP
//          DD DDNAME=SYSIN
//SYSLMOD DD LSN=*&LOAD,UNIT=&LDUNIT,SPACE=&LDROOM,DISP=&LDDISP,
//          DCB=BLKSIZE=6144
//SYSPRINT DD SYSOUT=*,DCB=(LRECL=121,BLKSIZE=1210)
//SYSUT1 DD SPACE=&E1ROOM,UNIT=(SASCR,SEP=(SYSLIN,SYSLMOD))
//GO EXEC PGM=*.EDT.SYSLMOD,COND=&GOIF,REGION=&GOREGN,
//          PARM='&GOPARM'
//FT05F001 DD DDNAME=SYSIN
//FT06F001 DD SYSOUT=*,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
```

Table C.15 IBM Cataloged Procedure FORTVCEG at ANL for Compile,
Link Edit, and Execution (Cont'd)

```
//FT07F001 DD  SYSOUT=B
//STEPLIB  DD  DISP=(OLD,PASS),DSN=*.EDT.SYSLMOD
//          DD  DISP=SHR,DSN=&GOLIB
```

Table C.16 IBM Cataloged Procedure FORTVCEP at ANL for Compile
and Link Edit

```

//FORTVCEP PROC AMDLIB='SYS1.AMDLIB',
//          AMDLIBX='SYS1.AMDLIBX',
//          COMPILE=FORTVS2,
//          EDITOR=IEWL,
//          EDTIF='(5,LT,FORTV)',
//          EDTOPTS=,
//          EDTREGN=512K,
//          E1ROOM='(CYL,(1,4))',
//          INCLUDE='SYS1.DUMMYLIB',
//          LIBRARY='SYS1.VSF2FORT',
//          LSIZE='(8192K,64K)',
//          LVL=77,
//          OBJBLK=3120,
//          OBJDISP='(OLD,DELETE)',
//          OBJECT='&OBJT',
//          OBJROOM='(CYL,(10,5))',
//          OBJUNIT=SASCR,
//          OPTIONS=,
//          PLOTUTL='SYS1.PLOTUTL',
//          POSTLIB='SYS1.DUMMYLIB',
//          PRELIB='SYS1.DUMMYLIB',
//          PRELIB2='SYS1.DUMMYLIB',
//          REGN=2048K,
//          STEPLIB='SYS1.VSF2COMP',
//          SYSTEM='*'
//FORTV     EXEC  PGM=&COMPILE,REGION=&REGN,
//               PARM='LANGLVL(&LVL),NOTF,&OPTIONS'
//STEPLIB   DD   DISP=SHR,DSN=&STEPLIB
//SYSLIB    DD   DISP=SHR,DSN=&INCLUDE
//SYSLIN    DD   UNIT=&OBJUNIT,DSNAME=&OBJECT,SPACE=&OBJROOM,
//               DISP=(MOD,PASS),DCB=(RECFM=FB,LRECL=80,BLKSIZE=&OBJBLK)
//SYSPRINT  DD   SYSOUT=*,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=3429)
//SYSPUNCH  DD   SYSOUT=B,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3440)
//SYSTEM    DD   SYSOUT=&SYSTEM,DCB=*.SYSPRINT
//EDT       EXEC  PGM=&EDITOR,PARM='DCBS,LIST,MAP,&EDTOPTS,SIZE=&LSIZE',
//               REGION=&EDTREGN,COND=&EDTIF
//SYSLIB    DD   DISP=SHR,DSN=&PRELIB
//          DD   DISP=SHR,DSN=&PRELIB2
//          DD   DISP=SHR,DSN=&LIBRARY
//          DD   DISP=SHR,DSN=&POSTLIB
//          DD   DISP=SHR,DSN=&PLOTUTL
//          DD   DISP=SHR,DSN=&AMDLIB
//          DD   DISP=SHR,DSN=&AMDLIBX
//SYSLIN    DD   DSNAME=*.FORTV.SYSLIN,DISP=&OBJDISP
//          DD   DDNAME=SYSIN
//SYSLMOD    DD   DDNAME=SYSPVT,DCB=BLKSIZE=6144
//SYSPRINT  DD   SYSOUT=*,DCB=(LRECL=121,BLKSIZE=1210)
//SYSUT1    DD   SPACE=&E1ROOM,UNIT=(SASCR,SEP=(SYSLIN,SYSLMOD))

```

Table C.17 IBM Cataloged Procedure RUN at ANL for Execution

```

//RUN      PROC B133=1596,   OR B133=133 TO DEBLOCK OUTPUT
//          DS=,
//          DSN=,
//          DUMP='YES,SYSOUT=F ', (DUMP TO FICHE) OR DUMP=NULLFILE
//          FT05=SYSIN,   OR FT05=FT05001 FOR PL1-GENERATED PROGRAMS
//          GOTIME=1439,
//          GOREGN=100K,
//          INDEX=' ',
//          L='.LOAD',
//          LDDISP='(SHR,PASS)',
//          M=G,
//          TIMEUP=NO322
//GO        EXEC PGM=&M,REGION=&GOREGN,ACCT=&TIMEUP,TIME=&GOTIME
//STEPLIB DD DISP=&LDDISP,DSN=&DS&DSN&INDEX&L
//FT05F001 DD DDNAME=&FT05
//FT06F001 DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=&B133)
//FT07F001 DD SYSOUT=B
//SYSPRINT DD SYSOUT=*,DCB=*.FT06F001
//SYSPNCH DD SYSOUT=B,DCB=(BLKSIZE=80,RECFM=F)
//SYSPUNCH DD SYSOUT=B,DCB=(BLKSIZE=80,RECFM=F)
//SYSUDUMP DD DSN=&DUMP

```

C.2 Execution

The following execution instructions assume that the COMMIX code system has been installed at your site as described in Sec. C.1. This involves creation of object libraries and making site-dependent changes in the sample jobstreams used for illustration. Contact the person at your site who installed the code to find out which code sections have been installed and to obtain the modified jobstreams. A short background section on code operation is given in Sec. C.2.1. Specific batch-mode execution examples are given in Sec. C.2.2.

C.2.1 Background

Under the recommended installation and operating procedure, the general user does not have access to the source code; instead, the user compiles (at least) one routine (see next two paragraphs) and links this with the appropriate COMMIX and system object files to form an executable code version during each job.

There are five programs in the COMMIX code system: the thermal-hydraulics code COMMIX and a set of four graphics postprocessor codes named VECT, TEMP, TDSC, and GRPH. All five of these programs use a quasidynamic storage allocation scheme for most arrays. Although the general user does not need to understand the details this storage scheme, the user must define the available size of the storage container. The user defines the available size of the storage container by specifying the dimension of array S in the COMMON BLOCK named SPACE in SUBROUTINE ALTER; the source code for this subroutine appears in each of the execution examples to be discussed below; the user modifies the line

```
COMMON /SPACE/ S(dimension),SPAEND
```

to specify the dimension using an INTEGER constant. Each job, then, consists of compiling SUBROUTINE ALTER, linking this with the rest of the code and system routines to form an executable code version, and then beginning execution. During execution, the code will compute and print the minimum storage container size needed based on your other input in the job and

compare this size with the dimension of the S array. If the S array is too small, the code will print an error message and stop; resubmit the job with array S dimensioned to the size indicated by the lines preceding the error message. The storage container size needed depends on your case (number of cells, thermal structures, etc.), typically ranging from a few thousand up to several hundred thousand; the storage container size needed for the four graphics postprocessors is smaller than that needed for the COMMIX code. The recommended approach is to run using the smallest S array dimension which will accommodate your case; under this approach, your jobs will use the minimum amount of memory, which is a factor in cost and turnaround time on many systems.

The user is provided the flexibility to override several routines in the COMMIX code. (Overrides are not needed in the four graphics postprocessor codes.) The three primary overrides allowed are for the specification of alternate forms for (1) specific heat capacity of structure materials using FUNCTION CPSOL1 (cf. Sec. 9.2), (2) thermal conductivity of structure materials using FUNCTION XKSOL1 (cf. Sec. 9.2), and (3) fluid/solid heat transfer coefficients using FUNCTION HEATCU (cf. Sec. 9.3); the entry point descriptions for these routines are discussed in the indicated sections of this report. Additionally, if COMMIX is to be run in interactive mode, you must supply a replacement for FUNCTION TLEFTS which returns a value which is greater than 10.0; an example of such a function is shown in Table C.2. (Regardless of whether the functionality of TLEFTS has been overridden or not, the code will stop if the value returned by TLEFTS is less than the value input using variable TREST in NAMELIST DATA (cf. Sec. 12.1).) An override is accomplished by you writing the source code for the routine(s) which you wish to override and including it in the jobstream as part of the file which contains the source code for SUBROUTINE ALTER.

Fluid number 1 in a COMMIX case may use either detailed or simplified properties; all other fluids must use simplified properties (cf. Sec. 9.1); therefore, only one set of detailed properties may be used in any given case. Four sets of detailed properties are distributed with the COMMIX code system: liquid sodium, liquid normal water, liquid heavy water, and helium gas. The person installing COMMIX must have created an object library for at least one set of detailed fluid properties; that person may have created object libraries for other fluids. The user

specifies which one of these sets is to be used during a COMMIX execution job by specifying a specific file name in the jobstream; the way in which this file name is specified is system dependent and is illustrated in Sec. C.2.2. Even if fluid number 1 is to use simplified properties, the file name identifying one of the sets of detailed fluid properties must be specified in the job in order to satisfy external entry point requirements.

Postprocessor codes VECT, TEMP, and GRPH call routines in the DISSPLA (Ref. C.4) package to produce graphical output on a file in device-independent DISSPOP metafile format. This file must be further postprocessed to produce viewable output on an actual graphical device (e.g. video terminal, plotter, film); the example jobstreams discussed in Sec. C.2.2 move the DISSPOP metafile to a PostScript-type laser printer using a locally written system script named "hardcopy" on the Cray and a locally written cataloged (system) procedure named "PPS" on the IBM. The person installing the COMMIX code system will have modified these example jobstreams to include proper references to the DISSPLA package and show you how to obtain viewable output at your site. If DISSPLA is not available at your installation, postprocessors VECT, TEMP, GRPH, and TDSC (TDSC prepares input for GRPH) were probably not installed.

Each of the five codes in the COMMIX code system (1) reads a card-image input file, (2) may read one or more binary input files supplied by you, (3) creates a printable output file having FORTRAN-type carriage-control information as the first character in each record, and (4) may create one or more binary output files for you to use in subsequent jobs. The card-image input is assumed to be in-stream in the examples. Naming conventions and other parameters associated with the binary files are suggested but are site dependent. Similarly, the information to be specified on the QSUB directive for the Cray and on the JOB card for the IBM are site dependent. The execution example jobstreams use the string "thisjob" to denote the job name; you should change this to a meaningful job name before submitting the job.

C.2.2 Specific Instructions

C.2.2.1 Cray

The example jobstreams for executing COMMIX and the four graphics postprocessor codes are shown in Tables C.5 through C.8. You should change the string "thisjob" to a meaningful job name for all jobs. You will additionally need to dimension array S in the in-stream source code for SUBROUTINE ALTER (which follows the "cat .. >SOURCE" command in the Prologue section), as discussed in Sec. C.2.1. The other changes needed are in the jobstream sections labeled "Prologue" and "Epilogue" and are discussed below for each type of execution job.

The example jobstream for executing COMMIX is shown in Table C.5. You select which one of the sets of detailed fluid properties is to be used during this job by completing the definition of shell variable CMXPROP in the Prologue section; allowed choices are shown in the example; you must specify one of the allowed choices even if fluid number 1 uses simplified properties. If you wish to override FUNCTIONS CPSOL1, XKSOL1, or HEATCU, include the appropriate FORTRAN-77 source code between the "END" card of SUBROUTINE ALTER and the "EOF1" in-stream terminator card in the "cat ... >SOURCE" part of the Prologue section. You must insert the card-image input in-stream following the "cat ... >CARDIN" command in the Prologue section; the format of this input is described in the main part of this report and is summarized in Appendix A. If this job is the continuation of a case from a previous job, then you must provide the name of the input restart file to be read in the "cp ... CMXOLD" command in the Prologue section; otherwise, remove this copy command. If you request that this job create a restart or a plot file, then these will be saved as "\$HOME/thisjob.r" and "\$HOME/thisjob.p", respectively; modify the "cp CMXNEW ..." and "cp PLTDAT ..." commands in the Epilogue section to save these files under different names.

The example jobstream for executing graphics postprocessors VECT and TEMP is shown in Table C.6. You select which one of these codes to run by completing the definition of shell variable CMXPLOT in the Prologue section, as noted in the example. You must insert the card-

image input in-stream following the "cat ... >CARDIN" command in the Prologue section; the format of this input is described in Appendix D.2 for VECT and Appendix D.3 for TEMP. The binary input file read by VECT and TEMP is, depending on the value of ITAPE in the card-image input file, either the restart or plot file created by a COMMIX execution job; you must specify an input file name in the "cp ... fort.9" command in the Prologue section. The person who installed COMMIX at your site will provide you with instructions concerning a replacement for "hardcopy" in the Epilogue section to send your graphics to a real device.

The example jobstream for executing graphics postprocessor TDSC is shown in Table C.7; this example assumes two consecutive executions of TDSC: a file concatenation followed by a data extraction. You must insert separate sets of card-image input in-stream in the Prologue section following the "cat ... >CARDIN1" command for the file concatenation (ITAPE>0) and following the "cat .. >CARDIN2" command for the data extraction (ITAPE=0); the format of this input is described in Appendix D.4. In concatenation mode, TDSC reads a series of ITAPE binary plot files which were created by a series of previous COMMIX execution jobs which are continuations of a single case; the user must supply ITAPE number of "cp ... fort.nn" commands in the Prologue section; on these copy commands, the values of "nn" must be used consecutively beginning with nn=21 and the corresponding input file names must present the data in proper chronological order (i.e. the file being copied onto file fort.21 contains data for case times which immediately precede the case times on the file being copied onto file fort.22). The binary file created during the concatenation step (which looks like a rather large COMMIX plot file) is passed internally to become the binary input file for the data extraction step. (You will not normally want to retain this large file past job completion; however, if you do wish to save the file then uncomment the "cp fort.21 ..." command in the Epilogue section.) The binary file created by the data extraction step is to be saved for use as input to the graphics postprocessor code GRPH; this is accomplished by the "cp fort.16 ..." command in the Epilogue section.

The example jobstream for executing graphics postprocessor GRPH is shown in Table C.8. You must insert the card-image input in-stream following the "cat ... >CARDIN" command in the Prologue section; the format of this input is described in Appendix D.5. The binary input

file read by GRPH is the binary file created by a TDSC data extraction run; you may use one primary input file and/or up to four alternate input files; you specify, in the Prologue section, the primary input file name (IFFT15=1) in the "cp ... fort.15" command and the alternate input file names (IFFT16>0) in the first IFFT16 of the "cp ... fort.nn" commands for nn=16, 12, 13, and 14; delete the copy commands for the files which are not being used. The person who installed COMMIX at your site will provide you with instructions concerning a replacement for "hardcopy" in the Epilogue section to send your graphics to a real device.

C.2.2.2 IBM

The sample jobstreams for executing COMMIX and the four graphics postprocessor codes are shown in Tables C.11 through C.14. These examples make use of cataloged procedures in use at ANL which differ substantially from those distributed by IBM; the person who installed the COMMIX code system at your site has probably made significant changes for your system and should have provided you with revised samples; however, the changes described below are typical of those you will have to make as a user. You should change the string "user_id_info" to a valid dataset name base index and the string "thisjob" to a meaningful job name for all jobs. All nontemporary datasets used in these jobs are assumed to be cataloged on disk. All printable output is assumed to be routed to the SYSOUT class specified by the MSGCLASS on the JOB card. You will additionally need to dimension array S in the in-stream source code for SUBROUTINE ALTER (which follows the DD card for DDNAME FORTV.SYSIN), as discussed in Sec. C.2.1. These jobstreams must be submitted as "unnumbered", since they will all contain NAMELIST-type in-stream input. The other changes needed are discussed below for each type of execution job.

The example jobstream for executing COMMIX is shown in Table C.11. If you wish to override FUNCTIONs CPSOL1, XKSOL1, or HEATCU, include the appropriate FORTRAN-77 source code as part of DDNAME FORTV.SYSIN between the "END" card of SUBROUTINE ALTER and the "/"* in-stream terminator card. You select which one of the sets of detailed fluid properties is to be used during this job by completing the DSN field on the DD card for DDNAME EDT.PROP; allowed choices are shown in the example; you must specify one of the

allowed choices even if fluid number 1 uses simplified properties. You must insert the card-image input in-stream following the DD card for DDNAME GO.SYSIN; the format of this input is described in the main part of this report and is summarized in Appendix A. If this job is the continuation of a case from a previous job, then you must provide the name of the input restart file to be read as the DSN field of the DD card for DDNAME GO.FT09F001; otherwise, remove this DD card. If you request that this job create a restart or a plot file, they will be saved based on the information specified on the DD cards for GO.FT10F001 and GO.FT76F001, respectively; you may need to alter the fields on these DD cards; in particular, the SPACE requirement is a function of the problem size for both files and also a function of the NTPLOT array for the plot file; if the output restart or plot file is not used, then remove the corresponding DD card from the jobstream.

The example jobstream for executing graphics postprocessors VECT and TEMP is shown in Table C.12. You select which one of these codes to run by completing (1) the DSN field on the DD card for DDNAME EDT.OBJECT and (2) the in-stream ENTRY linkage-editor directive in DDNAME EDT.SYSIN, as noted in the example. You must insert the card-image input in-stream following the DD card for DDNAME GO.SYSIN; the format of this input is described in Appendix D.2 for VECT and Appendix D.3 for TEMP. The binary input file read by VECT and TEMP is, depending on the value of ITAPE in the card-image input file, either the restart or plot file created by a COMMIX execution job; you must specify an input file name in the DSN field of the DD card for DDNAME GO.FT09F001. The SPACE parameter to be specified on the DD card for DDNAME GO.FT19F001 depends on the number and complexity of the figures being drawn. The person who installed COMMIX at your site will provide you with instructions concerning a replacement for the " EXEC PPS " step to send your graphics to a real device.

The example jobstream for executing graphics postprocessor TDSC is shown in Table C.13; this example assumes two consecutive executions of TDSC: a file concatenation followed by a data extraction. You must insert separate sets of card-image input in-stream following the DD card for DDNAME GO.SYSIN in (1) the CONCAT step for the file concatenation (ITAPE>0) and (2) the XTRACT step for the data extraction (ITAPE=0); the format of this input is

described in Appendix D.4. In concatenation mode, TDSC reads a series of ITAPE binary plot files which were created by a series of previous COMMIX execution jobs which are continuations of a single case; the user must supply ITAPE number of DD cards for DDNAMEs GO.FT09Fnnn in the CONCAT step; on these DD cards, the values of "nnn" must be used consecutively beginning with nnn=001 and the corresponding input file names must present the data in proper chronological order (i.e. the file assigned to DDNAME GO.FT09F001 contains data for case times which immediately precede the case times on the file assigned to DDNAME GO.FT09F002). The binary file created during the concatenation step (DDNAME GO.FT16F001 in the CONCAT step, which looks like a rather large COMMIX plot file) is passed internally to become the binary input file for the data extraction step (DDNAME GO.FT09F001 in the XTRACT step). You may need to alter the SPACE field on the DD card for DDNAME GO.FT16F001 in the CONCAT step depending on the amount of data being concatenated. (You will not normally want to retain this large file past job completion; however, if you do wish to save the file then (1) use a nontemporary dataset name in the DSN field (and modify the other fields) on the DD card for DDNAME GO.FT16F001 in the CONCAT step and (2) modify the DD card for DDNAME GO.FT09F001 in the XTRACT step to reference this file.) The binary file created by the data extraction step is to be saved for use as input to the graphics postprocessor code GRPH based on the information specified on the DD card for DDNAME GO.FT16F001 in the XTRACT step; you may need to alter the fields on this DD card; in particular, the SPACE requirement is a function of the amount of data extracted.

The example jobstream for executing graphics postprocessor GRPH is shown in Table C.14. You must insert the card-image input in-stream following the DD card for DDNAME GO.SYSIN; the format of this input is described in Appendix D.5. The binary input file read by GRPH is the binary file created by a TDSC data extraction run; you may use one primary input file and/or up to four alternate input files; you specify the primary input file name (IFFT15=1) in the DSN field of the DD card for DDNAME GO.FT15F001 and the alternate input file names (IFFT16>0) in the DSN field of the first IFFT16 of the DD cards for DDNAMEs GO.FTnnF001 for nn=16, 12, 13, and 14; delete the DD cards for the files which are not being used. The SPACE parameter to be specified on the DD card for DDNAME

GO.FT19F001 depends on the number and complexity of the figures being drawn. The person who installed COMMIX at your site will provide you with instructions concerning a replacement for the " EXEC PPS " step to send your graphics to a real device.

APPENDIX D: GRAPHICS POSTPROCESSORS

D.1 Introduction

The COMMIX program does not produce graphical output directly; instead, the program writes two binary files (plot and restart) which are then used as input to four postprocessor programs (VECT, TEMP, TDSC, and GRPH) to produce graphical output. The use of these files with these postprocessors is discussed in this appendix.

The plot file contains velocities and temperatures at time steps during the COMMIX run as specified in input array NTPLOT in NAMELIST DATA (cf. Sec. 12.6); this file is identified in the COMMIX execution sample jobstreams in Appendix C.2 by Cray file name PLTDAT and by IBM DDNAME GO.FT76F001. The restart file contains all variables at the last time step in a single COMMIX run if input variable IFRES in NAMELIST GEOM has the value 1, 3, or 4 (cf. Sec. 12.1); appended to this file are velocities and temperatures in "plot file format" for the last time step, regardless of the input in array NTPLOT; it is this "plot file appendage" section of the restart file which is read by the postprocessor programs; this file is identified in the COMMIX execution sample jobstreams in Appendix C.2 by Cray file name CMXNEW and by IBM DDNAME GO.FT10F001.

The four programs which postprocess these files to produce graphical output are summarized in Table D.1. Programs VECT and TEMP display results by superposition on two-dimensional planar cuts of the geometry grid for the case; these results are in the form of velocity vectors for VECT and temperature, as numerals or isolines, for TEMP; these two programs can read either the plot or restart file. Program TDSC prepares the binary input file to be input to GRPH; this is accomplished by reading a plot file and extracting a user-specified subset of information (velocities and temperatures at specific cell locations and specific times); this program may also be used to concatenate several plot files onto a single file. Program GRPH displays results in the form of curves showing velocities and temperatures as a function of time; this program can show multiple variables on each figure and can read data from up to five different binary files (useful for comparing results for several cases).

Table D.1 Graphics Postprocessor Programs

Program Name	Input File Type	Program Description
VECT	Plot Restart	Draws vectors superposed on planar cut of geometry to represent magnitude and direction of mass flow.
TEMP	Plot Restart	Draws numerals or isolines superposed on planar cut of geometry to represent temperature.
TDSC	Plot	Produces a binary file containing a user-selected subset of all velocities and temperatures reformatted for input to GRPH.
GRPH	TDSC	Draws plots of selected variables versus time.

In addition to the binary file input noted above, each postprocessor reads a card-image file, which contains processing directives (e.g. which planes and which variables to display); the remaining sections of this appendix describe the card-image input file requirements (including examples) for the four postprocessors; the majority of this input is in the form of NAMELISTs and is governed by the general rules covered in Sec. 2. The three programs which produce graphical output (VECT, TEMP, and GRPH) do so by making calls to routines in the DISSPLA software package. Further information about the DISSPLA package and sample jobstreams for all four postprocessors are given in Appendix C. Examples of the graphical output produced are presented as part of the discussion of the sample problem in Appendix E.

D.2 Program VECT

The VECT computer program produces a graphical display of the velocity field from a COMMIX calculation by using vectors superimposed on a planar view of the geometry grid. Each plot represents either an I plane displaying a y-z (or θ -z) grid, a J plane displaying a x-z (or r-z) grid, or a K plane displaying a x-y (or r- θ) grid. For each fluid cell (i.e. main control volume) within the selected plane, the VECT computer program calculates a cell-center velocity as the vector sum of the velocity components at the four cell edges which are cut by the planar view selected. This velocity is displayed as a vector originating from the cell center pointing in the direction of the cell-center velocity and having a length proportional to the magnitude of cell-center velocity (i.e. the speed).

The input guide for computer program VECT is shown in Table D.2. (The input guide is an integral part of the source coding; its printing may be selected by the user when executing the program; Table D.2 is a reproduction of this printing.) The first page of this input guide summarizes the files being used, location of sample jobstreams on computers at ANL, and the overall structure of the card-image input file. The card-image input file has one NAMELIST PLOT for each figure to be drawn. The remainder of this section discusses the input to be provided in NAMELIST PLOT; the discussion is ordered to follow the order in which the input variables are shown in the table.

The first eight variables (ITAPE through IREGI) shown in Table D.2 must be present in the first NAMELIST PLOT for their values to be effective; the values specified (or the defaults, if values are not explicitly input by the user) are in effect for this entire execution of program VECT; do not specify these variables in subsequent NAMELIST PLOT blocks. The value of ITAPE specifies the type of binary file which is providing the data from the COMMIX run; a value of 0 indicates this is a restart file (written by having IFRES=1, 3, or 4 in the COMMIX run) and a value greater than 0 indicates this is a plot file (written by having nonzero entries in the NTPLOT array in the COMMIX run). The value of IVER specifies which version of COMMIX wrote the binary input file; a value of 3 is appropriate for COMMIX-1AR/P. The value of IDOUBP specifies the level of precision used for floating-point numbers in the binary

file; the nominal choice is 1 for long-word machines and 2 for short-word machines. The value of **MENUE** controls whether this input guide will be printed; note that **MENUE=2** results in a printed copy of the input guide but no plots. The values of **XSCALE**, **YSCALE**, and **ZSCALE** are scale-factor multipliers for the I, J, and K directions, respectively; nonunity values would be used to distort the proportions of the display (for example, to make a particular coordinate direction more visible). The value of **IREG** specifies whether additional data for irregular cells are to be read from unit number 10. (The option selected by **IREGI>0** has not been used recently and is not documented; it may no longer be necessary to read this additional input; therefore, acceptance of the default (**IREGI=0**) is recommended.)

The next four variables (**BTIME** through **MOVIE**) are only used when **ITAPE>0**; they must be specified in the first **NAMelist PLOT** in order for the values specified to actually be used; they must not be specified in subsequent **NAMelist PLOT** blocks. When plotting from a plot (rather than restart) file, the user may choose to have the same plane(s) displayed at multiple times, in order to more easily view the changes during the case. (In contrast, planes may be displayed for only one time step when plotting from a restart file, since there are data for only one time step on a restart file.) The values specified for **BTIME**, **DTIME**, **ETIME**, and **MOVIE** control the frequency at which the plane(s) are displayed. For the case of **MOVIE=0**, the processing proceeds as follows: (1) the **VECT** program scans the plot file until it finds data for a case time which equals (or first exceeds) the value of **BTIME**; (2) the program then creates figures for all of the planes requested by the various **NAMelist PLOT** blocks; (3) the program then increments **BTIME** by **DTIME**; if the value of **BTIME** is less than the value of **ETIME**, then go back to step 1 (i.e. do the same planar views for the new time); if **BTIME** exceeds **ETIME**, then the job is complete. The values specified for **BTIME** and **ETIME** should be within the range of case times for which data are present on the plot file. The value specified for **MOVIE** alters the case times at which planar views are created: (1) the times selected under **MOVIE=0** are described above and are a subset of the case times for which data records exist on the plot file; if the case times on the plot file are spaced unevenly, the planar views will be spaced at uneven time increments (that will probably bear little relation to the value of **DTIME**); (2) specifying **MOVIE>0** produces a new planar view every **DTIME** seconds by performing a linear interpolation between adjacent case-time data records on the plot file; the

resulting planar views are, thus, evenly spaced in time. (The `MOVIE>0` option is most useful when preparing planar views to be viewed as a movie (either film or video tape); in this case, the user will only want to specify one `NAMelist PLOT`, so that only one planar view is repeated at sequentially increasing times; execute the `VECT` program a second time to produce a time sequence for another planar view.)

The remaining input variables may be specified in any `NAMelist PLOT` block; the value specified remains in effect until a new value is specified in a later `NAMelist PLOT` block.

Variables `IFIJK` through `KE` select the plane to be displayed. The value of `IFIJK` is the type of plane. The values specified for the two-element array `IJKL` select the specific plane number(s); there are two options. (1) A single plane is selected by specifying the plane number in array element `IJKL(1)` and a 0 for array element `IJKL(2)`. (2) If `IFIJK` is 1 or 2, two planes may be displayed on a single figure; the left side of the figure contains the mirror-image view of the plane number specified in array element `IJKL(1)`; the right side of the figure contains the normal-image view of the plane number specified (as a negative value) in array element `IJKL(2)`; this option is very useful when the cylindrical-coordinate fluid-grid geometry is used in `COMMIX`. Values may be specified for variables `IB`, `IE`, `JB`, `JE`, `KB`, and `KE` to restrict the view to being less than the full plane.

The planar views are placed on the "page" to make best use of the "available space"; the available space is based on an 8.5-inch by 11-inch logical page size having a 2-inch border at the bottom and a 1-inch border on the other three sides; the figure is centered within the available space. By default, the planar views are oriented so that the longest spatial dimension of the view runs in the direction of the longest dimension of the available space; this means that the program may choose either the 8.5-inch dimension or the 11-inch dimension of the page to be horizontal when viewing the figure; the user may override this default (and, thus, choose which dimension of the page is to be horizontal) by specifying a non-zero value for `IROTAT`. By default, the planar views are scaled (after orientation is decided) so that the view fills the available space; both dimensions are scaled by the same amount to preserve the proportionality

specified by XSCALE, YSCALE, and ZSCALE; a nonunity value may be specified for variable SIZE to enlarge (> 1) or reduce (< 1) this default figure size.

Boundary surfaces (both impervious and pervious) will always be shown using solid lines; the line width used for this may be altered using variable THK. By default, the faces between internal fluid cells (i.e. the coordinate grid lines) are not shown; these faces will be shown if the user specifies IGRID=1; a dotted line is used if the surface porosity is nonzero, and a solid line is used if the surface porosity is zero.

The value of IPERM selects the method to be used in computing the cell-center velocity components. These will be computed using

$$u_{i,ctr} = [(\gamma u)_{i,minus} + (\gamma u)_{i,plus}]/[2 \cdot \gamma_v] \quad (D.1)$$

for IPERM=1 and using

$$u_{i,ctr} = [(\gamma u)_{i,minus} + (\gamma u)_{i,plus}]/[\gamma_{i,minus} + \gamma_{i,plus} + 10^{-10}] \quad (D.2)$$

for IPERM=2, where u_i is the velocity component in direction i (i denoting symbolically either I, J, or K), γ_i is the surface porosity in direction i , γ_v is the volume porosity, and subscripts minus, plus, and ctr denote, respectively, the $-i$ face, the $+i$ face, and the cell center. The only difference between these two methods is the porosity value to use in the denominator; the two methods give the same results if porosity values are uniform.

The default value of ICONT (=0) must not be overridden. (This variable is reserved for future use when the VECT and TEMP computer programs are combined into a single program.) By default, the velocity vectors are scaled so that the longest vector does not get too long relative to the average grid spacing; the user will normally want to accept the default when first producing figures; the user may override the default scaling by specifying the scale factor, in units of meter/second per meter of grid, using variable SCALE; the value computed by the computer program for the scale factor is shown as "SCALET" in the printable output file to aid the user in selecting a value to override the default in a subsequent execution of program VECT; the user may want to explicitly specify the scale factor to result in a common velocity scaling among several different views. By default, the length of the vectors is linearly proportional to the magnitude of the velocity vector; the user may select logarithmic scaling (for which the

vector length is proportional to $\log_e(1 + \text{speed})$) by inputting `LSCALE > 0`. The style of vector used to represent the velocity is selected by input variable `IARROW`; this is a 4-digit integer code whose full meaning is discussed under parameter `IVEC` for routines `VECTOR` and `RLVEC` in Ref. 24; the default style of the velocity vector (`IARROW = 1131`) results in the vector starting at the cell center, a completely enclosed (but unshaded) arrow head is at the other end, and the size of the arrow head is the current character size. (In order to just get planar views of the geometry grid (without any velocity vectors), input the combination of `IARROW = 0` and `IGRID = 1`; this technique was used to produce Figs. E.1 through E.3.)

The bottom of the planar view will always be labeled to show the type of planar view, the plane number(s), a 1-inch long reference vector, and the speed (in [m/s] and [ft/s]) represented by this reference vector. If the input binary file is a restart file (`ITAPE = 0`), the user may have the case time added to this label by specifying `ITIME = 1`. (The case time is always shown when the input binary file is a plot file (`ITAPE > 0`).) The format used to display the value of the case time in this label area is selected using input variable `IDEC`. By default, the figure is enclosed by a box at the page-size limits and has a plot identification line (i.e. plot number, job name, user, date produced, etc.) in the margin; this border and plot identification are omitted if `ICLEAN = 1`. The user will normally want no debug output (`IBUG = 0`); the values of `IBUG > 0` are mainly for use during program development.

A sample card-image input file for program `VECT` is given in Table D.3. (Sample batch jobstreams for executing the `VECT` program are given in Tables C.6 (Cray) and C.12 (IBM).) This particular example is associated with the sample problem given in Appendix E. The first `NAMelist PLOT` specifies that the input binary file is a restart file (`ITAPE = 0`); velocities are to be displayed for J planes number 1 and number 3 (`IFIJK = 2` and `IJKL = 3, -1`); cell boundaries are to be shown (`IGRID = 1`); the time is to be displayed in exponent form (`ITIME = 1` and `IDEC = -2`). The resulting figure is shown as Fig. E.4. The second `NAMelist PLOT` specifies that velocities are to be displayed for K plane number 10 (`IFIJK = 3` and `IJKL = 10, 0`); unspecified variables retain their values from the previous `NAMelist PLOT`. The resulting figure is shown as Fig. E.5. Further discussion of the results shown in these two figures is given in Sec. E.4.

Table D.2 Input Guide for Program VECT

```
#####
####          Input Guide for VECT          ####
####          (Code updated 08/09/91)        ####
#####
##
##   This program reads a restart or plot file ##
## written by COMMIX and produces figures of   ##
## velocity vectors superposed on planar cross ##
## sections of the geometry grid.             ##
##                                              ##
#####
```

This program may be run on either the IBM or Cray;
however, the input binary file must be in the
format of the machine on which the code is running.

Files used by the program:

Binary Input, in COMMIX restart or plot file format:

IBM: FT09F001

Cray: fort.9

Binary Output, in DISSPLA DISSPOP metafile format:

IBM: FT19F001

Cray: popfil(DISSPLA10), popfil.dat(DISSPLA11)

Card Input, processing directives (see below):

IBM: FT05F001 (or GO.SYSIN)

Cray: "stdin"

Printed Output, with carriage control in col. 1:

IBM: FT06F001 Cray: "stdout"

Sample job streams may be found at ANL in:

IBM: B27484.COMMIX.IBMSAMP(PLOTVECT)

Cray(from IBM): B27484.COMMIX.CRSCRIPT(PVECTUSE)

Cray(from VAX): CC116:[B27484.CMXCRGEN]PVECTUSE.JOB

The input for each figure is given in NAMELIST PLOT in the form:

```
&PLOT VAR1=VALUE, VAR2=VALUE, ...      (Begin Fig.#1)
... ..
... .., VARN=VALUE, &END
&PLOT VAR1=VALUE, VAR2=VALUE, ...      (Begin Fig.#2)
... ..
... .., VARN=VALUE, &END
... .. etc. for other plots ... ..
```

Every line begins with at least one blank column.
The "&" preceding "PLOT" must be in column 2. For the
first figure, variables not specified assume defaults
shown in parentheses below. For subsequent figures,
variables not specified assume values used in preceding
figure. Certain variables must be specified in the
first NAMELIST PLOT (only) to be effective.

Table D.2 Input Guide for Program VECT (Cont'd)

== The Following Are Constant During a Plot Job ==
(Specify in the First NAMELIST PLOT Only)

- - - - - All Plot Jobs - - - - -

ITAPE Type of COMMIX file provided as input:
 = 0: Restart (i.e. FT10 or CMXNEW) (default)
 > 0: Plot (i.e. FT76 or PLTDAT)

IVER COMMIX version which wrote input file:
 = 1: -1, -1A
 = 2: -2
 = 3: -1AR/P (default)

IDOUBP Precision level of input file:
 = 1: Single (default; nominal Cray choice)
 = 2: Double (nominal IBM choice)

MENUE Print this Input Guide:
 = 0: No (but do requested plots) (default)
 = 1: Yes (and do requested plots)
 = 2: Yes and stop (no plots produced)

XSCALE = I-direction grid size multiplier (1.0)
YSCALE = J-direction grid size multiplier (1.0)
ZSCALE = K-direction grid size multiplier (1.0)

IREGI Read IREGular cell data from FT10:
 = 0: No (default)
 = 1: Yes (UNTRIED OPTION)

- - - Transient Plot Jobs (i.e. ITAPE>0) - - -

BTIME = Begin time of transient plots (0.0)
DTIME = Delta time of transient plots (1.0)
ETIME = End time of transient plots (BTIME)

Note: Given a transient plot file with data at a
sequence of times TIME-I, then code will:

1. Do all requested plots at first TIME-I which
satisfies: $BTIME \leq TIME-I \leq ETIME$.
2. Update BTIME: $BTIME = TIME-I + DTIME$.
3. If $BTIME \leq ETIME$, go to step 1; else end job.

MOVIE Movie option selection:
 = 0: Movie is not being made (default)
 > 0: Movie is being made:
 Must specify BTIME, DTIME, ETIME;
 DTIME=.04167 gives 24 frames/sec.

Table D.2 Input Guide for Program VECT (Cont'd)

== The Following May Change During a Plot Job ==
(Specify/Change in Any NAMELIST PLOT)

IFIJK Two-dimensional plane to be plotted:
 = 1: I-plane (y-z or theta-z)
 = 2: J-plane (x-z or r-z) (default)
 = 3: K-plane (x-y or r-theta)

IJKL Specific plane(s) to plot (default: 1,0):
 = M,0: Normal image of plane #M only
 = M,-N: Mirror image of plane #M on left and
 normal image of plane #N on right;
 only allowed for IFIJK=1 or 2.
 e.g. IJKL=1,0,: Plot plane #1 only
 IJKL=1,-6,: Image of #1 on left
 and plane #6 on right

IB = I-plane beginning index (1)
IE = I-plane ending index (IMAX)
JB = J-plane beginning index (1)
JE = J-plane ending index (JMAX)
KB = K-plane beginning index (1)
KE = K-plane ending index (KMAX)

SIZE = Scaling multiplier for plot size (1.0)

IROTAT Plot orientation on page:
 = 0: Chosen by code based on aspect ratio
 of figure (default)
 = 1: Force 11.0-inch side horizontal
 = 2: Force 8.5-inch side horizontal

IGRID Show internal cell boundaries:
 = 0: No (default)
 = 1: Yes, using dotted lines

THK = Line thickness for boundaries (0.02 inch)

IPERM = Porosity flag (1)
 (Input 2 if small porosities are used.)

ICONT Select variable to be plotted:
 = 0: Velocity vector (default)
 = 1 thru 4: Reserved for future use
 = 5 thru 7: Reserved for future use
 = 8 thru 9: Reserved for future use

SCALE Scale factor for velocity vector length:
 = 0.0: Calculated by code (default)
 > 0.0: User-specified value in units of
 meter/sec per meter of grid size.

LSCALE Type of velocity vector scaling:
 = 0: Linear (default)
 > 0: Logarithmic

Table D.2 Input Guide for Program VECT (Cont'd)

IARROW = Vector style and tip size (See DISSPLA
Manual: parameter IVEC as passed to
routines VECTOR/RLVEC; default=1131.)
= 0000: No vectors -- show geometry only

IBUG Debug print toggle:
= 0: No debug print (default; recommended)
= 1: A lot of extra output
= 2: Even more extra output

ITIME Show time on plot from restart file:
= 0: No (default; ignored if plot file input)
> 0: Yes (ignored if IARROW=0)

IDEC Display format for time (+1):
< 0: Exponent form, |IDEC| decimal places
> 0: Normal form, IDEC decimal places
> 100: Free form, IDEC-100 digits

ICLEAN Border and job identifier on figure:
= 0: Include (default)
= 1: Omit (ready for caption typing)

```
#####
##                               ##
##           End of Input Guide  ##
##                               ##
#####
```

Table D.3 Sample Card-Image Input File for Program VECT

```
&PLOT IFIJK=2, IJKL=3,-1, SCALE=0.0,
      ITAPE=0, IDOUBP=1, IGRID=1, ITIME=1, IDEC=-2, ICLEAN=1, &END
&PLOT IFIJK=3, IJKL=10,0, SCALE=0.0, &END
```


D.3 Program TEMP

The TEMP computer program produces a graphical display of the fluid temperature field from a COMMIX calculation by using digits or isotherms superimposed on a planar view of the geometry grid. Each plot represents either an I plane displaying a y-z (or θ -z) grid, a J plane displaying a x-z (or r-z) grid, or a K plane displaying a x-y (or r- θ) grid. The temperature is displayed in one of two formats: (1) the cell-center temperature is displayed for each fluid cell (i.e. main control volume) as a string of digits centered within the cell or (2) a series of dashed lines are drawn which represent isotherms. Isotherms are drawn by dividing the planar view of each fluid cell into several sections; within each section, the isotherms (if any) are positioned based on a multidirection interpolation between the temperature at the center of this cell and the temperature in the center of an adjacent cell (or an adjacent boundary surface element). The temperatures in thermal structures are not considered for either type of temperature display format.

The input guide for computer program TEMP is shown in Table D.4. (The input guide is an integral part of the source coding; its printing may be selected by the user when executing the program; Table D.4 is a reproduction of this printing.) The first page of this input guide summarizes the files being used, location of sample jobstreams on computers at ANL, and the overall structure of the card-image input file. The card-image input file has one NAMELIST PLOT for each figure to be drawn. The remainder of this section discusses the input to be provided in NAMELIST PLOT; the discussion is ordered to follow the order in which the input variables are shown in the table.

The first eight variables (ITAPE through IREGI) shown in Table D.4 must be present in the first NAMELIST PLOT for their values to be effective; the values specified (or the defaults, if values are not explicitly input by the user) are in effect for this entire execution of program TEMP; do not specify these variables in subsequent NAMELIST PLOT blocks. The value of ITAPE specifies the type of binary file which is providing the data from the COMMIX run; a value of 0 indicates this is a restart file (written by having IFRES=1, 3, or 4 in the COMMIX run) and a value greater than 0 indicates this is a plot file (written by having nonzero

entries in the NTPLOT array in the COMMIX run). The value of IVER specifies which version of COMMIX wrote the binary input file; a value of 3 is appropriate for COMMIX-1AR/P. The value of IDOUBP specifies the level of precision used for floating-point numbers in the binary file; the nominal choice is 1 for long-word machines and 2 for short-word machines. The value of MENUE controls whether this input guide will be printed; note that MENUE=2 results in a printed copy of the input guide but no plots. The values of XSCALE, YSCALE, and ZSCALE are scale-factor multipliers for the I, J, and K directions, respectively; nonunity values would be used to distort the proportions of the display (for example, to make a particular coordinate direction more visible). The value of IREG specifies whether additional data for irregular cells are to be read from unit number 10. (The option selected by IREGI>0 has not been used recently and is not documented; it may no longer be necessary to read this additional input; therefore, acceptance of the default (IREGI=0) is recommended.)

The next four variables (BTIME through MOVIE) are only used when ITAPE>0; they must be specified in the first NAMELIST PLOT in order for the values specified to actually be used; they must not be specified in subsequent NAMELIST PLOT blocks. When plotting from a plot (rather than restart) file, the user may choose to have the same plane(s) displayed at multiple times, in order to more easily view the changes during the case. (In contrast, planes may be displayed for only one time step when plotting from a restart file, since there are data for only one time step on a restart file.) The values specified for BTIME, DTIME, ETIME, and MOVIE control the frequency at which the plane(s) are displayed. For the case of MOVIE=0, the processing proceeds as follows: (1) the TEMP program scans the plot file until it finds data for a case time which equals (or first exceeds) the value of BTIME; (2) the program then creates figures for all of the planes requested by the various NAMELIST PLOT blocks; (3) the program then increments BTIME by DTIME; if the value of BTIME is less than the value of ETIME, then go back to step 1 (i.e. do the same planar views for the new time); if BTIME exceeds ETIME, then the job is complete. The values specified for BTIME and ETIME should be within the range of case times for which data are present on the plot file. The value specified for MOVIE alters the case times at which planar views are created: (1) the times selected under MOVIE=0 are described above and are a subset of the case times for which data records exist on the plot file; if the case times on the plot file are spaced unevenly, the planar

views will be spaced at uneven time increments (that will probably bear little relation to the value of DTIME); (2) specifying MOVIE > 0 produces a new planar view every DTIME seconds by performing a linear interpolation between adjacent case-time data records on the plot file; the resulting planar views are, thus, evenly spaced in time. (The MOVIE > 0 option is most useful when preparing planar views to be viewed as a movie (either film or video tape); in this case, the user will only want to specify one NAMELIST PLOT, so that only one planar view is repeated at sequentially increasing times; execute the TEMP program a second time to produce a time sequence for another planar view.)

The remaining input variables may be specified in any NAMELIST PLOT block; the value specified remains in effect until a new value is specified in a later NAMELIST PLOT block.

Variables IFIJK through KE select the plane to be displayed. The value of IFIJK is the type of plane. The values specified for the two-element array IJKL select the specific plane number(s); there are two options. (1) A single plane is selected by specifying the plane number in array element IJKL(1) and a 0 for array element IJKL(2). (2) If IFIJK is 1 or 2, two planes may be displayed on a single figure; the left side of the figure contains the mirror-image view of the plane number specified in array element IJKL(1); the right side of the figure contains the normal-image view of the plane number specified (as a negative value) in array element IJKL(2); this option is very useful when the cylindrical-coordinate fluid-grid geometry is used in COMMIX. Values may be specified for variables IB, IE, JB, JE, KB, and KE to restrict the view to being less than the full plane.

The planar views are placed on the "page" to make best use of the "available space"; the available space is based on an 8.5-inch by 11-inch logical page size having a 2-inch border at the bottom and a 1-inch border on the other three sides; the figure is centered within the available space. By default, the planar views are oriented so that the longest spatial dimension of the view runs in the direction of the longest dimension of the available space; this means that the program may choose either the 8.5-inch dimension or the 11-inch dimension of the page to be horizontal when viewing the figure; the user may override this default (and, thus, choose

which dimension of the page is to be horizontal) by specifying a non-zero value for IROTAT. By default, the planar views are scaled (after orientation is decided) so that the view fills the available space; both dimensions are scaled by the same amount to preserve the proportionality specified by XSCALE, YSCALE, and ZSCALE; a nonunity value may be specified for variable SIZE to enlarge (> 1) or reduce (< 1) this default figure size.

Boundary surfaces (both impervious and pervious) will always be shown using solid lines; the line width used for this may be altered using variable THK. By default, the faces between internal fluid cells (i.e. the coordinate grid lines) are not shown; these faces will be shown if the user specifies IGRID=1; a dotted line is used if the surface porosity is nonzero, and a solid line is used if the surface porosity is zero.

The default value of ICONT (=1) must not be overridden. (This variable is reserved for future use when the TEMP and VECT computer programs are combined into a single program.) The value input for IUNITS selects the units ($[^{\circ}\text{C}]$, $[^{\circ}\text{F}]$, or $[\text{K}]$) to be used for displaying temperature values. The value of IDIGIT selects between the two display formats. (1) The isoline format is selected by IDIGIT=0; when this format is selected, the user should specify values for the minimum temperature isotherm to be displayed (TMIN), the maximum temperature isotherm to be displayed (TMAX), and the temperature increment between adjacent isotherms (DELTA); note that the values specified for TMIN, TMAX, and DELTA are all in the units selected by IUNITS; the value of LABEL specifies how minimum and maximum temperature points are to be labeled; this labeling will be done using a character height of HITE*SIZE inches. (2) The digit format is selected by IDIGIT $\neq 0$; the temperature values will be shown as integers if IDIGIT > 0 and as floating-point numbers having $|\text{IDIGIT}|$ digits to the right of the decimal point if IDIGIT < 0 ; the digits will be displayed using a character height of HITE*SIZE inches and the angle between the horizontal and the base of the character will be PHI degrees; the special value of PHI=999 is often convenient for displays showing the $r-\theta$ plane.

The bottom of the planar view will always be labeled to show the type of planar view, the plane number(s), and the temperature units. If the input binary file is a restart file

(ITAPE=0), the user may have the case time added to this label by specifying ITIME=1. (The case time is always shown when the input binary file is a plot file (ITAPE>0).) The format used to display the value of the case time in this label area is selected using input variable IDEC. By default, the figure is enclosed by a box at the page-size limits and has a plot identification line (i.e. plot number, job name, user, date produced, etc.) in the margin; this border and plot identification are omitted if ICLEAN=1. The user will normally want no debug output (IBUG=0); the values of IBUG>0 are mainly used during program development and can produce rather enormous amounts of printed output.

A sample card-image input file for program TEMP is given in Table D.5. (Sample batch jobstreams for executing the TEMP program are given in Tables C.6 (Cray) and C.12 (IBM).) This particular example is associated with the sample problem given in Appendix E. The first NAMELIST PLOT specifies that the input binary file is a restart file (ITAPE=0); temperatures are to be displayed for J planes number 1 and number 3 (IFIJK=2 and IJKL=3, -1); the display format is isotherms spaced every 10 °C between 30 °C and 550 °C (IDIGIT=0, DELT=10., TMIN=30., and TMAX=550. (and IUNITS=0 by default)); cell boundaries are to be shown (IGRID=1); the time is to be displayed in exponent form (ITIME=1 and IDEC=-2). The resulting figure is shown as Fig. E.6. The second NAMELIST PLOT specifies that temperatures are to be displayed for K plane number 9 (IFIJK=3 and IJKL=9,0); the display format is digits as integers with the special θ -based orientation (IDIGIT=1 and PHI=999.); unspecified variables retain their values from the previous NAMELIST PLOT. The resulting figure is shown as Fig. E.7. Further discussion of the results shown in these two figures is given in Sec. E.4

Table D.4 Input Guide for Program TEMP

```
#####
####      Input Guide for TEMP      ####
####      (Code updated 08/08/91)    ####
#####
##
##   This program reads a restart or plot file  ##
## written by COMMIX and produces figures of    ##
## temperature superposed on planar cross      ##
## sections of the geometry grid. This is      ##
## done by drawing isotherm lines or by showing ##
## a number in each grid cell.                 ##
##                                              ##
#####
```

This program may be run on either the IBM or Cray; however, the input binary file must be in the format of the machine on which the code is running.

Files used by the program:

Binary Input, in COMMIX restart or plot file format:

IBM: FT09F001

Cray: fort.9

Binary Output, in DISSPLA DISSPOP metafile format:

IBM: FT19F001

Cray: popfil(DISSPLA10), popfil.dat(DISSPLA11)

Card Input, processing directives (see below):

IBM: FT05F001 (or GO.SYSIN)

Cray: "stdin"

Printed Output, with carriage control in col. 1:

IBM: FT06F001

Cray: "stdout"

Sample job streams may be found at ANL in:

IBM: B27484.COMMIX.IBMSAMP(PLOTTEMP)

Cray(from IBM): B27484.COMMIX.CRSCRIPT(PTEMPUSE)

Cray(from VAX): CC116:[B27484.CMXCRGEN]PTEMPUSE.JOB

The input for each figure is given in NAMELIST PLOT in the form:

```
&PLOT VAR1=VALUE, VAR2=VALUE, ...      (Begin Fig.#1)
... ..
... .., VARN=VALUE, &END
&PLOT VAR1=VALUE, VAR2=VALUE, ...      (Begin Fig.#2)
... ..
... .., VARN=VALUE, &END
... .. etc. for other plots ... ..
```

Every line begins with at least one blank column. The "&" preceding "PLOT" must be in column 2. For the first figure, variables not specified assume defaults shown in parentheses below. For subsequent figures, variables not specified assume values used in preceding figure. Certain variables must be specified in the first NAMELIST PLOT (only) to be effective.

Table D.4 Input Guide for Program TEMP (Cont'd)

== The Following Are Constant During a Plot Job ==
(Specify in the First NAMELIST PLOT Only)

- - - - - All Plot Jobs - - - - -

ITAPE Type of COMMIX file provided as input:
 = 0: Restart (i.e. FT10 or CMXNEW) (default)
 > 0: Plot (i.e. FT76 or PLTDAT)

IVER COMMIX version which wrote input file:
 = 1: -1, -1A
 = 2: -2
 = 3: -1AR/P (default)

IDOUBP Precision level of input file:
 = 1: Single (default; nominal Cray choice)
 = 2: Double (nominal IBM choice)

MENUE Print this Input Guide:
 = 0: No (but do requested plots) (default)
 = 1: Yes (and do requested plots)
 = 2: Yes and stop (no plots produced)

XSCALE = I-direction grid size multiplier (1.0)
YSCALE = J-direction grid size multiplier (1.0)
ZSCALE = K-direction grid size multiplier (1.0)

IREGI Read IREGular cell data from FT10:
 = 0: No (default)
 = 1: Yes (UNTRIED OPTION)

- - - Transient Plot Jobs (i.e. ITAPE>0) - - -

BTIME = Begin time of transient plots (0.0)
DTIME = Delta time of transient plots (1.0)
ETIME = End time of transient plots (BTIME)

Note: Given a transient plot file with data at a
sequence of times TIME-I, then code will:

1. Do all requested plots at first TIME-I which
satisfies: $BTIME \leq TIME-I \leq ETIME$.
2. Update BTIME: $BTIME = BTIME + DTIME$.
3. If $BTIME \leq ETIME$, go to step 1; else end job.

MOVIE Movie option selection:
 = 0: Movie is not being made (default)
 > 0: Movie is being made:
 Must specify BTIME, DTIME, ETIME;
 DTIME=.04167 gives 24 frames/sec.

Table D.4 Input Guide for Program TEMP (Cont'd)

== The Following May Change During a Plot Job ==
(Specify/Change in Any NAMELIST PLOT)

IFIJK Two-dimensional plane to be plotted:
 = 1: I-plane (y-z or theta-z)
 = 2: J-plane (x-z or r-z) (default)
 = 3: K-plane (x-y or r-theta)

IJKL(2) Specific plane(s) to plot (default: 1,0):
 = M,0: Normal image of plane #M only
 = M,-N: Mirror image of plane #M on left and
 normal image of plane #N on right;
 only allowed for IFIJK=1 or 2.
 e.g. IJKL=1,0,: Plot plane #1 only
 IJKL=1,-6,: Image of #1 on left
 and plane #6 on right

IB = I-plane beginning index (1)
IE = I-plane ending index (IMAX)
JB = J-plane beginning index (1)
JE = J-plane ending index (JMAX)
KB = K-plane beginning index (1)
KE = K-plane ending index (KMAX)

SIZE = Scaling multiplier for plot size (1.0)

IROTAT Plot orientation on page:
 = 0: Chosen by code based on aspect ratio
 of figure (default)
 = 1: Force 11.0-inch side horizontal
 = 2: Force 8.5-inch side horizontal

IGRID Show internal cell boundaries:
 = 0: No (default)
 = 1: Yes, using dotted lines

THK = Line thickness for boundaries (0.02 inch)

ICON Select variable to be plotted:
 = 0: Reserved for future use
 = 1: Temperature (default)
 = 2 thru 4: Reserved for future use
 = 5: Mass fractions #1 (COMMIX-2)
 = 6: Mass fractions #2 (COMMIX-2)
 = 7: Mass fractions #3 (COMMIX-2)
 = 8 thru 9: Reserved for future use

IDIGIT Display mode for variable selected (0):
 = 0: Isolines (see TMIN,TMAX,DELT,LABEL)
 != 0: By individual cell (see HITE,PHI) as
 >0: INTEGER
 <0: REAL: |IDIGIT| decimal places

IUNITS Units for temperature plots (ICON=1):
 = 1: Degrees Celsius (default)
 = 2: Degrees Fahrenheit (1.8*C + 32.)
 = 3: kelvins (C + 273.15)

TMIN = Minimum value of isoline (10.0 in IUNITS)
TMAX = Maximum value of isoline (300.0 in IUNITS)
DELT = Isoline spacing (10.0 in IUNITS)

Table D.4 Input Guide for Program TEMP (Cont'd)

LABEL Labeling of min/max on isoline plot (2):
 = 0: None
 = 1: Near cell center
 = 2: Outside with line to cell center

HITE = Height of numerals at SIZE=1. (0.07 inch)
PHI = Angle of numerals w.r.t. horiz. (0.0 deg.)
 (=999.: Angle is perpendicular to theta)

IBUG Debug print toggle:
 = 0: No debug print (default; recommended)
 = 1: A lot of extra output
 = 2 or 3: Even more extra output

ITIME Show time on plot from restart file:
 = 0: No (default; ignored if plot file input)
 > 0: Yes (always shown if plot file input)

IDEC Display format for time (+1):
 < 0: Exponent form, |IDEC| decimal places
 > 0: Normal form, IDEC decimal places
 > 100: Free form, IDEC-100 digits

ICLEAN Border and job identifier on figure:
 = 0: Include (default)
 = 1: Omit (ready for caption typing)

```
#####
##                                     ##
##           End of Input Guide       ##
##                                     ##
#####
```

Table D.5 Sample Card-Image Input File for Program TEMP

```
&PLOT IFIJK=2, IJKL=3,-1,
      IDIGIT=0, TMIN=30., TMAX=550., DELT=10., LABEL=2,
      ITAPE=0, IDOUBP=1, IGRID=1, ITIME=1, IDEC=-2, ICLEAN=1, &END
&PLOT IFIJK=3, IJKL=9,0, IDIGIT=1, PHI=999., &END
```

D.4 Program TDSC

The TDSC computer program is used to prepare the binary file required for input to the GRPH computer program. There are two processing modes in the TDSC program: (1) data extraction and (2) file concatenation. In the data-extraction mode, the TDSC program selects a user-specified subset of the velocities and temperatures from a plot file which was written by the COMMIX computer program and writes these data in the binary file format expected by the GRPH program. The file-concatenation mode is used to combine two or more plot files written by the COMMIX program into a single file which has the same binary format as the input plot files. Typically, the binary file produced by execution of the TDSC program in the file-concatenation mode will be used as input to an execution of the TDSC program in the data-extraction mode.

The input guide for computer program TDSC is shown in Table D.6. (The input guide is an integral part of the source coding; its printing may be selected by the user when executing the program; Table D.6 is a reproduction of this printing.) The first page of this input guide summarizes the files being used, location of sample jobstreams on computers at ANL, and the overall structure of the card-image input file. The card-image input file consists of one NAMELIST SET. The remainder of this section discusses the input to be provided in NAMELIST SET; the discussion is ordered to follow the order in which the input variables are shown in the table.

The first ten variables (MENUE through IOLD) shown in Table D.6 must be specified for both data-extraction and file-concatenation modes of operation. The value of MENUE controls whether this input guide will be printed; note that MENUE=2 results in a printed copy of the input guide but no files are processed. The value of ITAPE specifies the operating mode; a value of 0 indicates the data-extraction mode; specifying ITAPE>0 indicates the file-concatenation mode and the value of ITAPE is the number of files to be concatenated. The frequency at which data records are written on the output binary file is governed by NBEG, NEND, and NTPRNT; the data for a time step will be written on the output file if the time step number is (1) between NBEG and NEND and (2) an integral multiple of NTPRNT; note that

the "time step number" as used here is the time step number (NT) as shown in the execution of the COMMIX program, not the ordinal position of time step records on the input binary plot file.

By default, the time step numbers for which data are written on the binary output file are noted in the printable output file produced by the TDSC program; if this is a data-extraction mode execution, the actual data values are also listed in the printable output file; this information is not shown in the printable output file if IPRINT=1. The value of IVER specifies which version of COMMIX wrote the binary input file; a value of 1 is appropriate for COMMIX-1AR/P. The value of IDOUBP specifies the level of precision used for floating-point numbers in the binary input file; the nominal choice is 1 for long-word machines and 2 for short-word machines. The precision of floating-point numbers in the output binary file depends on the execution mode: (1) the output file has the same precision as the input binary file for the file-concatenation mode and (2) the output file has single-precision values for the data-extraction mode. The user will normally want no debug output (IBUG=0); the values of IBUG > 0 are mainly used during program development and can produce rather enormous amounts of printed output. The user should not override the default value (=0) of IOLD; the TDSC program automatically processes header records.

The remaining two variables are only specified when executing the TDSC program in the data-extraction mode. The values input in arrays ITHPR and NTHPR determine which specific variables are to be written on the output binary file. (In contrast, the values of all variables are written on the binary output file during an execution of the TDSC program in file-concatenation mode.) The value of array element ITHPR(i) specifies the type of variable (e.g. velocity, temperature, or heat source) and the corresponding value of array element NTHPR(i) specifies a specific spatial location. The various allowable values for these two arrays are shown in Table D.6. The spatial location code in array element NTHPR(i) depends on the variable type code in array element ITHPR(i): (1) for a cell-center quantity (i.e. fluid temperature and volumetric heat source), the location code is a six-digit integer of the form "iijjkk", where "ii", "jj", and "kk" are two-digit integers specifying, respectively, the I, J, and K indices of the fluid cell; (2) for a cell-edge quantity (i.e. velocity components), the same six-digit format is used and

selects the value on the "plus" (not "minus") side of the cell; (3) for a boundary-surface-element quantity (i.e. velocity and temperature), an eight-digit integer of the form "nnijjkk" is used, where "nn" is a two-digit integer specifying the boundary surface number and "ijjkk" is a six-digit integer specifying the fluid cell location which is immediately adjacent to this boundary surface element and pointed to by the boundary surface unit-normal vector; (4) for thermal structure temperatures, the value specified is the storage location in array TTS (which, unfortunately, is not an easy value to specify properly (cf. Sec. 4.7 of Ref. 14). The values of variables are written on the output binary file in the order in which the variable codes are entered in arrays ITHPR and NTHPR; when using the GRPH program to subsequently process this binary file, the variables to be plotted will be referenced (via input variables NVAR1 through NVAR5 and array N16VAR) by their position on the binary file as assigned during the execution of the TDSC program. Up to 100 variables may be selected (a limitation imposed by array dimensions in the GRPH program); if more than 100 variables are needed then perform an additional data-extraction mode execution of the TDSC program which writes the additional variables to a different binary output file.

Sample card-image input files for program TDSC are shown in Table D.7. (Sample batch jobstreams for executing the TDSC program are given in Tables C.7 (Cray) and C.13 (IBM).) This particular example is associated with the sample problem given in Appendix E. Part (a) of the table is the input needed to concatenate five plot files (ITAPE=5); data records are written to the output binary file (in plot file format) for all time steps (NTPRNT=1) between time steps number 1 (=NBEG) and number 319 (=NEND); for each time step, the output data records contain values for all variables. (Since the sample problem in Appendix E attained steady-state in a single computer run, a file-concatenation-mode execution of the TDSC program was not actually needed.) Part (b) of the table is the input needed for a (subsequent) data-extraction mode execution (ITAPE=0); data records are written to the output binary file (in the format for input to the GRPH program) for all time steps between time steps number 1 (=NBEG) and number 319 (=NEND); since "NTPLOT=-10" was input when the sample problem was executed on the COMMIX-1AR/P program, data exist on the plot file for time step numbers which are an integral multiple of ten (i.e. for time step numbers 10, 20, ..., and 310). For each of these time steps, the TDSC program is to extract

the values of ten variables and write them to the output binary file; the first three variables are the axial velocity (ITHPR(i)=3 for i=1,2,3) at the +z face of the following fluid cell locations (NTHPR(i) for i=1,2,3): (2,2,7) (sodium in pump), (3,2,5) (argon at mid-height just outside reactor vessel (RV)), and (4,2,5) (argon at mid-height just inside guard vessel (GV)); the last seven variables are the temperature (ITHPR(i)=4 for i=4,...,10) for the following fluid cell locations (NTHPR(i) for i=4,...,10): (1,2,4) (sodium at core inlet), (1,2,5) (sodium at core exit), (3,2,1) (argon adjacent to RV at bottom of RV/GV gap), (3,2,11) (argon adjacent to RV at top of RV/GV gap), (5,2,11) (air adjacent to GV at top of GV/CV gap), (6,2,11) (air adjacent to cavity vessel (CV) at top of GV/CV gap), and (8,3,10) (sodium at exit of intermediate side of intermediate heat exchanger (IHX)). The binary output file created by this data-extraction mode execution of the TDSC program will be used as the binary input file for the GRPH program and provide a basis for the sample GRPH program execution which is discussed in Sec. D.5. (As a reminder, the executions of the TDSC program do not produce any graphical output.)

Table D.6 Input Guide for Program TDSC

```
#####
####          Input Guide for TDSC          ####
####          (Code updated 08/09/91)        ####
#####
##
##   This program manipulates plot files from ##
##   COMMIX (i.e. FT76F001 or PLTDAT). Primary ##
##   purpose is to extract selected data and ##
##   write it in a form readable by the "GRPH" ##
##   code, which plots variables vs. time.    ##
##   Secondary purpose is to concatenate several ##
##   COMMIX plot files into a single file for ##
##   input to the above extraction process.   ##
##                                             ##
#####
```

This program may be run on either the IBM or Cray;
however, the input/output binary files must be in the
format of the machine on which the code is running.

Files used by the program:

Binary Input, in COMMIX plot file format:

IBM: FT09F001, FT09F002, FT09F003, ..., FT09Fnnn

Cray: fort.21, fort.22, fort.23, ..., fort.nn

Binary Output, will be in "PLOT" or "GRPH" format:

IBM: FT16F001

Cray: fort.16

Card Input, processing directives (see below):

IBM: FT05F001 (or GO.SYSIN)

Cray: "stdin"

Printed Output, with carriage control in col. 1:

IBM: FT06F001 Cray: "stdout"

Sample job streams may be found at ANL in:

IBM: B27484.COMMIX.IBMSAMP(PLOTTDSC)

Cray(from IBM): B27484.COMMIX.CRSCRIPT(PTDSCUSE)

Cray(from VAX): CC116:[B27484.CMXCRGEN]PTDSCUSE.JOB

The processing directives are input as NAMELIST SET,
which has the following general format:

```
&SET VARIABLE1=VALUE, VARIABLE2=VALUE, ...,
..., ..., VARIABLEN=VALUE, &END
```

where the initial "&" must be in column 2, column 1 is
always blank, no embedded blanks in variable names or
values, and a comma after every value; variables not
input take defaults shown in parentheses below.

Table D.6 Input Guide for Program TDSC (Cont'd)

The following variables may be input in any order :

MENUE Print this Input Guide:
 = 0: No (but do process files) (default)
 = 1: Yes (and do process files)
 = 2: Yes and stop (no files processed)

ITAPE Mode of operation:
 = 0: Extraction: Read 1 input file and
 write output file in "GRPH" format
 containing variables selected by ITHPR
 and NTHPR. (default)
 = 1: Copy: Read 1 input file and write
 output file in "PLOT" format containing
 all data. (rarely used)
 = N: Concatenation: Read N (>1) input
 files and write output file in "PLOT"
 format containing all variables. Data
 from file M is appended to data from
 files 1 thru M-1 during this process.
 (Output file is always governed by NTPRNT, etc.)

The following three variables control the frequency
 at which data is written to the output file. (The
 "timestep" refers to the timestep number in the
 COMMIX case -- not the record number on the file.)

NBEG = No writes before this timestep (1)
 NEND = No writes after this timestep (99999)
 NTPRNT = Between NBEG and NEND, write if timestep
 is integral multiple of NTPRNT. (1)

IPRINT Print data written to output file:
 = 0: Write timestep and time; if extracting,
 also write data values. (default)
 = 1: Omit these edits.

IVER COMMIX version which wrote input file:
 = 1: -1, -1A, -1AR/P (default)
 = 2: -2

IDOUBP Precision level of input file:
 = 1: Single (default; nominal Cray choice)
 = 2: Double (nominal IEM choice)
 (Output File Precision: Same as input file during
 concatenation; always single during extraction.)

IBUG Select debug prints:
 = 0: Off (default)
 = 1: On

IOLD Write header on output file:
 = 0: Yes (default; nominal choice)
 = 1: No (obsolete option)
 (Input 0 for file concatenation; after first file
 is processed, header writing is suppressed.)

- - End of Input for Concatenation Mode (ITAPE>0) - -

Table D.6 Input Guide for Program TDSC (Cont'd)

- - - - Data Extraction Mode (ITAPE=0) Only - - - -

The next two arrays select the data and its ordering on the output file.

ITHPR(N) Type of the N-th variable

NTHPR(N) Grid location of the N-th variable

Up to 100 values may be specified in each array;
the following table shows proper specifications:

ITHPR	NTHPR	Variable Selected
01	IIJJKK	UL: x(r)-dir. cell velocity
02	IIJJKK	VL: y(theta)-dir. cell velocity
03	IIJJKK	WL: z-dir. cell velocity
04	IIJJKK	TL: cell temperature
05	NNIIJJKK	TLB: boundary surface temperature
06	NNIIJJKK	VELBN: boundary surface velocity
07	IIJJKK	QSOUR: cell volumetric heat source
10	M	TTS: thermal structure temperature

In the above table:

II, JJ, and KK : two-digit cell indices,

NN : boundary surface number, and

M : thermal structure element number (TRICKY).

```
#####
##                               ##
##           End of Input Guide   ##
##                               ##
#####
```

Table D.7 Sample Card-Image Input Files for Program TDSC

(a) File-Concatenation Mode:

```
&SET  ITAPE=5, NBEG=1, NTPRNT=1, NEND=319, IDOUBP=1,  &END
```

(b) Data-Extraction Mode:

```
&SET
  ITAPE=0, NBEG=1, NTPRNT=1, NEND=319, IDOUBP=1,
  ITHPR=3*3, 7*4,
  NTHPR=020207,030205,040205,
        010204,010205,030201,030211,050211,060211,080310,
&END
```


D.5 Program GRPH

The GRPH computer program produces graphical displays of variables from a COMMIX calculation as plots of variable values versus time. Each plot may display up to nine variables: up to five from a "primary" binary input file and one each from up to four "alternate" binary input files. The capability for multiple binary input files is provided, primarily, to allow showing results for several COMMIX calculations on the same plot; however, the program may be used to display more than five results from a primary calculation and/or more than one result from an alternate calculation if one (or more) of these binary input files is actually a copy of one of the other binary input files.

The input guide for computer program GRPH is shown in Table D.8. (The input guide is an integral part of the source coding; its printing may be selected by the user when executing the program; Table D.8 is a reproduction of this printing.) The first page of this input guide summarizes the files being used, location of sample jobstreams on computers at ANL, and the overall structure of the card-image input file. Unlike the other three programs discussed in this appendix, the card-image input file for the GRPH program consists of more than just NAMELIST-type input records.

The first record in the card-image input file contains three integers in I6 format (i.e. each number must be right-justified in a six-character field). The first field indicates whether the primary binary input file is used; this file is used if $IFFT15=1$ and is not used if $IFFT15=0$. The second field indicates whether the alternate binary input files are used; if $IFFT16>0$ then these files are used and the value of $IFFT16$ is the number (≤ 4) of alternate files; if $IFFT16=0$ then no alternate files are used. At least one of $IFFT15$ and $IFFT16$ must be nonzero; both may be nonzero. In either case, the input binary files are the output binary files from data-extraction-mode executions of the TDSC program. The third field (MENUE) controls whether this input guide is printed; note that specifying a value of 2 here results in a printed copy of the input guide but no plots.

The remainder of the card-image input file consists of two character-type records and a NAMELIST PLOT for each figure to be drawn. The first of these two character-type records (identified as "Card #2" in Table D.8) specifies the title line to be used above the plot. The second of these two character-type records (identified as "Card #3" in Table D.8) specifies the axis labeling string for the dependent variable, which is plotted along the Y (vertical) axis. Each of these records may contain up to 60 characters and should be terminated with a dollar sign ("\$\$") to insure proper processing of the string; the case (upper and lower) of all input letters will be preserved by the GRPH program in drawing these two strings on the figure. The independent variable (time) is plotted along the X (horizontal) axis, which will be labeled "TIME (s)". The remainder of the input for each figure is provided in NAMELIST PLOT; the following discussion is ordered to follow the order in which the input variables are shown in Table D.8.

The first six variables (IMULT through COMIXB) under NAMELIST PLOT are associated with the data to be plotted from the primary binary input file; their values should be entered for IFFT15=1. The value of IMULT indicates how many variables (≤ 5) are to be plotted; if IMULT=0 then no variables are to be plotted from the primary binary input file. The specific variables to be plotted are identified by the values input in variables NVAR_i, for $1 \leq i \leq \text{IMULT}$; the values input for variables NVAR_i are the sequential storage order of the variables on the binary input file as defined during the execution of the TDSC program in the data-extraction mode (and are, thus, less than or equal to 100); input NVAR_i=0 if no variable is to be plotted for curve number i. As shown in Table D.8, each of the five NVAR_i curves has its own line type for ease of identification on the plot. (Nominally the first IMULT of variables NVAR_i should be nonzero; however, there are some exceptions allowed; for example, if the user wants to plot only a single variable but use a dashed line, input that variable's storage order as variable NVAR3 along with IMULT=3, NAVR1=0, and NVAR2=0.) By default, each curve represents a different variable; the optional specification of NAVG12=1 (with IMULT=1) allows drawing a single solid line representing the average of variables NVAR1 and NAVR2. By default, the values plotted are the values as read from the binary input file; optional input allows the value read from the binary input file to be modified by a multiplicative (COMIXA) and/or additive (COMIXB) constant; the values of COMIXA and COMIXB apply to all of the IMULT curves for the primary binary input file on this figure.

The next four variables (N16VAR through EXPERB) under NAMELIST PLOT are associated with the data to be plotted from the alternate binary input files; their values should be entered for $IFFT16 > 0$. The specific variable to be plotted from alternate file number n is identified by the value input in array element $N16VAR(n)$ for $1 \leq n \leq IFFT16$; the variable is identified by its sequential storage order on the binary input file as defined during the execution of the TDSC program in the data-extraction mode; input $N16VAR(n)=0$ if no variable is to be plotted from alternate file number n . As shown in Table D.8, the value input in array element $IMARK(n)$ determines the line type and the symbol spacing to be associated with the data plotted from alternate file number n ; each file's data may be represented by just a line (e.g. $IMARK(1)=20000$), just the symbols (e.g. $IMARK(1)=25$), or the symbols and a line (e.g. $IMARK(1)=20025$). (The type of symbol to be used in identifying curves is fixed by the alternate file number n : a box ($\square$) for $n=1$, a triangle (Δ) for $n=2$, a diamond ($\diamond$) for $n=3$, and an inverted triangle (∇) for $n=4$.) By default, the values plotted are the values as read from the binary input file; optional input allows the values read from alternate binary input file number n to be modified by a multiplicative (array element $EXPERA(n)$) and/or additive (array element $EXPERB(n)$) constant.

The next eleven variables (XORG through LYAX2) are associated with the two axes on the plot. For the X axis (which is time), the user specifies the beginning value (XORG), the ending value (XMAX), the increment value for labeling axis values (XSTEP), the number of tick marks per XSTEP along the axis (IXTICK), and the axis length (XAXIS); data values which are outside of the range bounded by XORG and XMAX will not be plotted. The corresponding values for the Y axis are specified using variables YORG, YMAX, YSTEP, IYTICK, and YAXIS; if $YORG=YMAX=0$ then the GRPH program will set YORG, YMAX, and YSTEP based on the range of the values being plotted; data values which are outside of the range bounded by YORG and YMAX will not be plotted (and a warning to this effect will appear in the printable output file). An auxiliary Y axis, showing an alternate temperature scale, will be drawn on the right-hand side of the plot if a nonzero value is input for LYAX2. As with the other graphics programs, the logical page size is 8.5 inch by 11 inch; the GRPH program will center the plot within the available space, which is defined to be the page size less a 2-inch border at the bottom and a 1-inch border on the other three sides. By default, the X axis will

be drawn parallel to the 11-inch length of the page; specifying $IROTAT=2$ will force the X axis to be parallel to the 8.5-inch side of the page. (The user must consider the page orientation selected by $IROTAT$ when choosing axis lengths; unlike the $VECT$ and $TEMP$ programs, the $GRPH$ program does not rescale the figure to fit the available space; if the maximum axis length recommendations in Table D.8 are exceeded, part of the plot may be clipped.)

By default, each curve on a figure is drawn by connecting adjacent points with a straight line; specifying $INTERP>0$ selects one of three methods for smoothing the curve to be drawn; the value specified for $INTERP$ applies to all curves on the figure. The thickness of the line used to draw the curves may be altered using input variable THK ; the value specified is (1) inches if $THK<1$ and (2) a multiplier on the default line thickness if $THK>1$. The data values will be listed on the printable output file (as well as being plotted) if $IPRINT>0$. By default, the figure is enclosed by a box at the page-size limits and has a plot identification line (i.e. plot number, job name, user, date produced, etc.) in the margin; this border and plot identification are omitted if $ICLEAN=1$.

A sample card-image input file for program $GRPH$ is given in Table D.9. (Sample batch jobstreams for executing the $GRPH$ program are given in Tables C.8 (Cray) and C.14 (IBM).) This particular example is associated with the sample problem given in Appendix E with data extracted from the plot file using the input in Table D.7b; since this was not a transient calculation, the resulting figures will merely show the progression toward a converged steady-state solution. The first record specifies that only the primary binary input file is used ($IFFT15=1$ and $IFFT16=0$). Six figures are requested. The first figure will show one ($IMULT=1$) curve: curve number 1 is variable number 1 ($NVAR1=1$) from the binary input file, which is the axial velocity at the +z face of fluid cell (2,2,7) (i.e. pump); the X (time) axis is restricted to the range 0 s ($=XORG$) through 10^4 s ($=XMAX$) (and, thus, only displays the first 220 time steps of the approach to steady state); the Y axis is restricted to the range 16.0 m/s ($=YORG$) to 16.6 m/s ($=YMAX$); the plot is oriented with the X axis parallel to the 8.5-inch side of the page ($IROTAT=2$); the axis lengths ($XAXIS$ and $YAXIS$) are the recommended maximum values for this plot orientation; the resulting plot is shown in Fig. E.8a. The second figure will show two curves: curve number 1 is variable number 2 ($NVAR1=2$),

which is the axial velocity at the +z face of fluid cell (3,2,5) (i.e. argon at mid-height adjacent to reactor vessel (RV)), and curve number 2 is variable number 3 (NVAR2=3), which is the axial velocity at the +z face of fluid cell (4,2,5) (i.e. argon at mid-height adjacent to guard vessel (GV)); the Y axis limits are to be computed by the GRPH program (since YORG=YMAX=0); the X axis limits, both axis lengths, and the orientation are the same as for the first figure since they are not specified in the second NAMELIST PLOT; the resulting plot is shown in Fig. E.8b. The third figure will show two curves: curve number 1 is variable number 4 (NVAR1=4), which is the temperature of fluid cell (1,2,4) (i.e. sodium at core inlet), and curve number 2 is variable number 5 (NVAR2=5), which is the temperature of fluid cell (1,2,5) (i.e. sodium at core exit); the resulting plot is shown in Fig. E.9a. The fourth figure will show one curve: curve number 1 is variable number 10 (NVAR1=10), which is the temperature of fluid cell (8,3,10) (i.e. sodium at exit of intermediate side of intermediate heat exchanger (IHX)); Y axis limits are specified; the resulting plot is shown in Fig. E.9b. The fifth figure will show two curves: curve number 1 is variable number 6 (NVAR1=6), which is the temperature of fluid cell (3,2,1) (i.e. argon adjacent to RV at bottom of RV/GV gap), and curve number 2 is variable number 7 (NVAR2=7), which is the temperature of fluid cell (3,2,11) (i.e. argon adjacent to RV at top of RV/GV gap); the resulting plot is shown in Fig. E.9c. The sixth figure will show two curves: curve number 1 is variable number 8 (NVAR1=8), which is the temperature of fluid cell (5,2,11) (i.e. air adjacent to GV at top of GV/CV gap), and curve number 2 is variable number 9 (NVAR2=9), which is the temperature of fluid cell (6,2,11) (i.e. air adjacent to cavity vessel (CV) at top of GV/CV gap); the resulting plot is shown in Fig. E.9d. Further discussion of the results shown in these six plots is given in Sec. E.4.

Table D.8 Input Guide for Program GRPH

```
#####
####      Input Guide for GRPH      ####
####      (Code updated 10/10/91)    ####
#####
##
##   This program plots COMMIX results vs. time   ##
## by reading binary input file(s) prepared by   ##
## the "TDSC" program, which reads the plot file ##
## written by COMMIX and extracts only user-     ##
## selected data.                                ##
##                                                ##
#####
```

This program may be run on either the IBM or Cray; however, the input/output binary files must be in the format of the machine on which the code is running.

Files used by the program:
 Binary Input, as prepared by TDSC:
 Primary: IBM: FT15F001 Cray: fort.15
 Alternates: IBM: FT16F001, FT12F001, FT13F001, FT14F001
 Alternates: Cray: fort.16, fort.12, fort.13, fort.14
 Binary Output, in DISSPLA DISSPOP metafile format:
 IBM: FT19F001
 Cray: popfil(DISSPLA10), popfil.dat(DISSPLA11)
 Card Input, processing directives (see below):
 IBM: FT05F001 (or GO.SYSIN)
 Cray: "stdin"
 Printed Output, with carriage control in col. 1:
 IBM: FT06F001 Cray: "stdout"

Sample batch job streams may be found at ANL in:
 IBM: B27484.COMMIX.IBMSAMP(PLOTGRPH)
 Cray(from IBM): B27484.COMMIX.CRSCRIPT(PGRPHUSE)
 Cray(from VAX): CC116:[B27484.CMXCRGEN]PGRPHUSE.JOB

The processing directives are input as follows:

```
Card #1                               (Only once per job)
Card #2                               (Begin figure #1)
Card #3
  &PLOT VAR1=VAL, VAR2=VAL, ...
  ..., VAR3=VAL, &END
Card #2                               (Begin figure #2)
Card #3
  &PLOT VAR1=VAL, VAR2=VAL, ...
  ..., VAR3=VAL, &END
...(Repeat #2,#3 and &PLOT for all figures desired)...
```

The description of these cards follows:

Table D.8 Input Guide for Program GRPH (Cont'd)

Card #1: IFFT15, IFFT16, MENUE (FORMAT(3I6))
 IFFT15 = 0: Primary input file is not used
 = 1: Primary input file is used
 IFFT16 = 0: Alternate input files are not used
 > 0: Number (<=4) of Alternate input files used
 (IFFT15 and IFFT16 may not both be zero.)
 MENUE = 0: Do not print input guide
 = 1: Print input guide
 = 2: Print input guide and stop (no plots)

Card #2: Figure top title (FORMAT(A60))
 (60 char.max.; must end with "\$"; up/low case)

Card #3: Y-axis label (FORMAT(A60))
 (60 char.max.; must end with "\$"; up/low case)
 (The X-axis label is "TIME (s)")

&PLOT: Variables in NAMELIST format
 Column 1 of every card is blank; the initial "&" must
 be in column 2; no embedded blanks in variable names
 or values; a comma after each value; variables not
 input take the default values shown in parentheses
 below on the first plot; variables not entered for
 subsequent plots retain values from the prior plot.
 The allowed variables are as follows:

IMULT = # of variables plotted per figure from Primary
 file (default = 1; minimum = 0; maximum = 5)

Each curve uses a different line pattern and
 variables NVAR1 through NVAR5 select specific data
 to be plotted from Primary file; input sequential
 storage order as defined during the TDSC run; must
 input NVARi for 1<=i<=IMULT when IMULT>0;
 if NVARi=0 the corresponding curve is not drawn.
 (Default: NVARi=0 for 1<=i<=IMULT.)

NVAR1 = First curve, solid line (IMULT>0)
 NVAR2 = Second curve, dotted line (IMULT>1)
 NVAR3 = Third curve, dashed line (IMULT>2)
 NVAR4 = Fourth curve, chain-dotted line (IMULT>3)
 NVAR5 = Fifth curve, chain-dashed line (IMULT>4)

NAVG12 = 0: No special averaging is done (default)
 > 0: 1st curve drawn from Primary file will be
 the average of variables NVAR1 & NVAR2;
 NAVG12>0 requires NVAR1>0 and NVAR2>0.
 (If IMULT>1, the curves selected by NVAR2
 through NVAR5 will also be drawn.)

COMIXA = Constant to modify Primary data (1.0)
 COMIXB = Constant to modify Primary data (0.0)
 Plotted Value = (COMIXA * DATA + COMIXB)

Table D.8 Input Guide for Program GRPH (Cont'd)

If plotting from Alternate file(s) (IFFT16>0), input values in the following four arrays for each Alternate file number n, for 1<=n<=IFFT16.

N16VAR(n) = Data variable number to be plotted from Alternate file n. Input sequential storage order as defined during the TDSC run. (0)

IMARK(n) = 10000*(line type)+(symbol spacing) for data from Alternate file n, where
 (line type): 0=no line, 1=solid, 2=dot, 3=dash, 4=chaindot, 5=chaindash; and
 (symbol spacing): 0=no symbols, >0=number of data points between symbols.
 (Defaults: 20025, 30025, 40025, 50025.)

EXPERA(n) = Constant to modify Alternate file n data (1.0)
 EXPERB(n) = Constant to modify Alternate file n data (0.0)
 Plotted Value = EXPERA(n) * DATA + EXPERB(n)

XORG = Beginning time of graph X-axis
 XSTEP = Time increment of graph X-axis
 XMAX = Maximum time of graph X-axis
 XAXIS = X-axis length (8.0 inch)
 IXTICK = # of tick marks per XSTEP along X-axis (2)

YORG = Minimum function value on Y-axis (0.)
 YSTEP = Function increment of graph Y-axis (0.)
 YMAX = Maximum function value on Y-axis (0.)
 (If YORG=YMAX=0. code will calculate YORG, YSTEP, and YMAX from the data being plotted.)
 YAXIS = Y-axis length (4.2 inch)
 IYTICK = # of tick marks per YSTEP along Y-axis (1)

LYAX2 Allows an auxiliary Y-axis to be drawn on right-hand side of figure with limits scaled from YORG, YSTEP, and YMAX:
 = 0: No auxiliary axis (default)
 = 1: Convert from DEG-C to DEG-F
 = 2: Convert from DEG-F to DEG-C

IROTAT Orientation of X-axis on page:
 = 1: Parallel to 11.0-inch side (default)
 (Recommend: XAXIS<=8.0, YAXIS<=4.2.)
 = 2: Parallel to 8.5-inch side
 (Recommend: XAXIS<=5.5, YAXIS<=6.7.)

INTERP Method for interpolation during curve drawing:
 = 0: Linear (default)
 = 1: 3rd-order polynomial (uses PARA3)
 = 2: 5th-order polynomial (uses PARA5)
 = 3: 3rd-order rational spline (uses RASPLN(3))

THK = Curve thickness (2.0)

Table D.8 Input Guide for Program GRPH (Cont'd)

IPRINT Print data points on FT06F001:
 = 0: No (default)
 > 0: Yes

ICLEAN Border and job identifier on figure:
 = 0: Include (default)
 = 1: Omit (ready for caption typing)

```
#####
#####          #####
##### End of Input Guide #####
#####          #####
#####
```

Table D.9 Sample Card-Image Input File for Program GRPH

```

      1      0      0
(a) Pump: WL(2,2,7)$
VELOCITY (m/s)$
&PLOT
  IMULT=1, NVAR1=1,NVAR2=0,NVAR3=0,NVAR4=0,NVAR5=0,
  XORG=0., XSTEP=2.0E+3, XMAX=1.0E+4,
  YORG=16., YSTEP=0.2, YMAX=16.6,
  IROTAT=2, XAXIS=5.5, YAXIS=6.7, ICLEAN=1,
&END
(b) Argon: WL(3,2,5), WL(4,2,5)$
VELOCITY (m/s)$
&PLOT
  IMULT=2, NVAR1=2,NVAR2=3,NVAR3=0,NVAR4=0,NVAR5=0,
  YORG=0., YMAX=0.,
&END
(a) Core In/Out: TL(1,2,4), TL(1,2,5)$
TEMPERATURE (C)$
&PLOT
  IMULT=2, NVAR1=4,NVAR2=5,NVAR3=0,NVAR4=0,NVAR5=0,
  YORG=0., YMAX=0.,
&END
(b) IHX-int Exit: TL(8,3,10)$
TEMPERATURE (C)$
&PLOT
  IMULT=1, NVAR1=10,NVAR2=0,NVAR3=0,NVAR4=0,NVAR5=0,
  YORG=390., YSTEP=10., YMAX=430.,
&END
(c) Argon Bot/Top: TL(3,2,1), TL(3,2,11)$
TEMPERATURE (C)$
&PLOT
  IMULT=2, NVAR1=6,NVAR2=7,NVAR3=0,NVAR4=0,NVAR5=0,
  YORG=200., YSTEP=100., YMAX=500.,
&END
(d) Air Exit: TL(5,2,11), TL(6,2,11)$
TEMPERATURE (C)$
&PLOT
  IMULT=2, NVAR1=8,NVAR2=9,NVAR3=0,NVAR4=0,NVAR5=0,
  YORG=65., YSTEP=5., YMAX=95.,
&END

```

APPENDIX E: SAMPLE PROBLEM

This appendix uses a sample problem to illustrate the preparation of the card-image input file and the format of the printed output file for the COMMIX-1AR/P program. This particular problem, although small in number of cells, uses many of the model options in the computer program. In addition to the required features, such as establishing the geometry and boundary conditions, the problem includes such options as multiple fluids, thermal structures, radiation heat transfer, force structures, expansion cell velocity boundary conditions, and a pump.

The problem is described in Sec. E.1, the card-image input file is described in Sec. E.2, and selected portions of the printed output are shown in Sec. E.3. Examples of the graphical output produced by postprocessor programs VECT, TEMP, and GRPH (which are described in Appendix D) are shown in Sec. E.4. Examples of the card-image input files needed for continuations of this case are given in Sec. E.5.

E.1 Description

The sample problem is a steady-state calculation of the conditions for a pool-type liquid-metal-cooled nuclear reactor system. The model will consider the reactor vessel cavity but not the intermediate or steam loops which are external to the cavity. The specific values mentioned below do not correspond to any one specific design.

The main portion of the modeling covers the interior of the reactor vessel (RV). The steel RV is cylindrical, having an outer radius of 2.87 m and a wall thickness of 50.8 mm; the coolant, liquid sodium, fills the RV to a height of 12.68 m above the bottom. Within the RV, coolant flow is as follows, beginning at the cold pool in the lower portion of the RV: up through the fuel assemblies to the hot pool in the upper portion of the RV, downward through the intermediate heat exchangers (IHX), upward through the pumps, and then downward through pipes to return to the cold pool. The fuel assemblies are in the central portion of the RV, and the other components are on the periphery of the RV. The symmetrical arrangement of components allows the modeling to be accomplished using a 90° sector.

Each fuel assembly consist of 217 fuel pins (3.315 mm outer radius and 0.457 mm thickness of stainless-steel cladding) arranged on a triangular pitch (8.088 mm) within a hexagonal assembly wall; the coolant flow area is $5.0 \times 10^{-3} \text{ m}^2$ per assembly and the hydraulic diameter is 3.39 mm. The overall length of the fuel pins is 4.41 m; the bottom end of the fuel pins is 1.17 m above the bottom of the RV; the actual fuel region is 1.73 m long, begins 1.26 m above the bottom of the fuel pin, and is a metallic-based material. The modeling of the fuel assemblies will be on a lumped, rather than discretized basis; the 90°-sector model contains 4079 fuel pins generating a total of 106.25 MW.

The IHX is a counter-flow straight-tube design, with the RV sodium flowing downward in the tubes and the intermediate-loop sodium flowing upward on the shell side. The stainless-steel tubes have an outside radius of 9.525 mm, a wall thickness of 0.978 mm, and a length of 5.34 m; the bottom of the tubes coincides with the top of the fuel pins in axial elevation. There are 1000 such tubes in the 90°-sector model.

There is one pump in the 90°-sector model. Its rated conditions are a volumetric flow rate of $0.64368 \text{ m}^3/\text{s}$, a head of 84.7 m (0.726 MPa), a speed of 1116 rev/min, and a torque of $0.026981 \text{ MN} \cdot \text{m}$. The coolant flow area within the pump is 0.04223 m^2 . The pipe which directs the sodium flow from the pump outlet to the cold pool is cylindrical and has an inner radius of 0.202 m.

The RV is sealed (i.e. no open inlet or outlet path for the primary system sodium), and the pressure at the top of the sodium is atmospheric. The temperature of the cold pool sodium is 320 °C. At the rated full-power/full-flow conditions, the temperature difference between the cold and hot pools is 150 °C.

The RV is surrounded by a cylindrical guard vessel (GV). The steel GV has an outer radius of 3.023 m and a wall thickness of 50.8 mm. The gap between the RV and the GV is filled with argon gas at atmospheric pressure and sealed (i.e. no open inlet or exit path for the argon).

The GV is surrounded by a cylindrical cavity vessel (CV). The steel CV has an outer radius of 3.353 m and a wall thickness of 25.4 mm. The gap between the GV and the CV is filled with air; the bottom and top of this gap are connected by separate ducts to a region above the RV to provide a path for the natural circulation of atmospheric air through the gap; this provides a passive cooling system for the RV at decay-heat power levels. For the purposes of this sample problem, the ductwork will not be modeled; a velocity boundary condition at the bottom of the gap will be specified which provides an air mass flow rate of 4.3 kg/s for the 90°-sector model.

The only part of the intermediate cooling loop modeled is the sodium flow through the shell-side of the IHX. The flow area associated with 1000 tubes is 0.4869 m². The intermediate loop sodium has a mass flow rate of 568 kg/s and enters the bottom of the IHX at a temperature of 282 °C.

E.2 Card-Image Input File

The card-image input file for the sample problem is shown in Table E.1.

All records up to the record which begins with " &GEOM " are the Problem Description input block. All records in this block are comments which provide a summary description of the problem, showing allocations of fluid cells, boundary surfaces, thermal structures, force structures, etc.

The remaining subsections explain various portions of the card-image input file which follow the Problem Description input block. The material is presented by model category following the order in which material is presented in Secs. 3 through 12 of the main part of this report, rather than progressing from the beginning to the end of the card-image input file.

Table E.1 Listing of Card-Image Input File

```

Mini-mod (8,3,12): 90-degree sector of modular pool-type LMR:
  Includes code+IHX(pri)+pump+IHX(int)+RVACS but not to scale.
  Steady-State: full-power(106MW), full-flow.
Flow path for sodium within reactor vessel(RV) is (I,K,J):
  up thru fuel assemblies (1,1-3,4-6) with heat source in K=5
  to hot pool (1,1-3,7-9)+(1-2,1-3,10-11)
  down thru IHX (2,3,3-9) with heat transfer in K=7-9
  up thru pump (2,2,3-9) with DELTA-P supplied at K=7
  down thru pipe (2,1,3-9)
  to cold pool (1-2,1-3,1-2)+(1,1-3,3)
  to fuel assemblies (closed loop).
Stagnant argon between RV and guard vessel(GV) (3-4,1-3,1-11).
Flowing air between GV and confinement vessel(CV) (5-6,1-3,1-11).
Radiation heat transfer between RV, GV, and CV.
(No cells in radial node I=7.)
Intermediate side of IHX in (8,3,6-10) with heat transfer in K=7-9.
Expansion Cells: primary-sodium=(2,1,12), argon=(4,1,12).
Boundary Surfs.: 1-8=pri.Na; 9-15=argon; 16-21=air; 22-27=int.Na;
  28-33=zero.porosity.internals.
Thermal Structs.: 1=fuel; 2=IHX; 3=RV; 4=GV; 5=CV.
Fluids: #1=sodium(pri+int); #2=argon; #3=air.
Solids: #1=fuel(metal); #2=steel.
Force Structs.: 1-3=core; 4=IHX(pri); 5=IHX(int); 6=pipe; 7=pump.
Heat Transf. Correls.: 1=core; 2=unused; 3=IHX(pri); 4=IHX(int);
  5=RV/GV(gap); 6=GV/CV(gap); 7=RV(in);
Trans. Functs.: 1=core power; 2=pump torque.
Turbulence: user-input constant.
S array: 71749 on short-word, 64405 on long-word machine.
Revised 02/18/92, P.Garner, PLGarner@anl.gov, 708-252-4872
&GEOM
  IGEOM=-1, NL1=482, NM1=205,
  ISYMCH=5, IFITEN=3, IFHTX=3, IFREB=0, IFRES=1,
  ITURKE=0, NFORCE=7, ISTRUC=2, ISTBUG=0, IBSBUG=1,
  IMAX=8, JMAX=3, KMAX=12, NSURF=33,
  DX=1.55,1.30,2*.076,2*.171,.161,.750,
  DY=2*.3927,.7854,
  DZ=2*.15,.51,1.26,1.73,1.42,3*1.78,2*1.06,1.0,
  XNORML=1.,-1.,4*0., 0.,-1., 1.,-1.,5*0., 1.,-1.,4*0.,
    1.,-1.,4*0., 1.,-1.,4*0.,
  YNORML=2*0.,1.,-1.,2*0., 2*0., 2*0.,1.,-1.,3*0., 2*0.,1.,-1.,2*0.,
    2*0.,1.,-1.,2*0., 2*0.,1.,-1.,2*0.,
  ZNORML=4*0.,1.,-1., -1.,0., 4*0.,1.,2*-1., 4*0.,1.,-1.,
    4*0.,1.,-1., 4*0.,1.,-1.,
&END
REG      -1.    1    1    1    3    1   11    1  RV:R+(CTR)
REG      -1.    2    2    1    1   12   12    1  RV(EXP):R+
REG      -1.    2    2    1    2    3    9    2  RV:R-(NO.Q)
REG      -1.    2    2    1    1   12   12    2  RV(EXP):R-(NO.Q)
REG      -1.    1    1    1    1    1    3    3  RV:O+
REG      -1.    1    1    1    1    7   11    3  RV:O+
REG      -1.    2    2    1    1    1    1    3  RV:O+
REG      -1.    2    2    1    1   10   11    3  RV:O+
REG      -1.    2    2    1    1   12   12    3  RV(EXP):O+
REG      -1.    1    1    3    3    1    3    4  RV:O-
REG      -1.    1    1    3    3    7   11    4  RV:O-
REG      -1.    2    2    3    3    1    6    4  RV:O-
REG      -1.    2    2    3    3   10   11    4  RV:O-
REG      -1.    2    2    1    1   12   12    4  RV(EXP):O-

```

Table E.1 Listing of Card-Image Input File (Cont'd)

REG	-1.	1	2	1	3	1	1	5	RV:Z+
REG	-1.	1	1	1	3	11	11	6	RV:Z-
REG	-1.	2	2	2	3	11	11	6	RV:Z-
REG	-1.	2	2	1	1	12	12	7	RV(EXP):Z-
REG	-1.	2	2	1	3	1	2	8	RV:R-(Q)
REG	-1.	2	2	3	3	3	9	8	RV:R-(Q)
REG	-1.	2	2	1	3	10	11	8	RV:R-(Q)
REG	-1.	3	3	1	3	1	11	9	RV/GV:R+
REG	-1.	4	4	1	1	12	12	9	RV/GV:R+(EXP)
REG	-1.	4	4	1	3	1	11	10	RV/GV:R-
REG	-1.	4	4	1	1	12	12	10	RV/GV:R-(EXP)
REG	-1.	3	4	1	1	1	11	11	RV/GV:O+
REG	-1.	4	4	1	1	12	12	11	RV/GV:O+(EXP)
REG	-1.	3	4	3	3	1	11	12	RV/GV:O-
REG	-1.	4	4	1	1	12	12	12	RV/GV:O-(EXP)
REG	-1.	3	4	1	3	1	1	13	RV/GV:Z+(NOFLOW)
REG	-1.	3	3	1	3	11	11	14	RV/GV:Z-(NOFLOW)
REG	-1.	4	4	2	3	11	11	14	RV/GV:Z-(NOFLOW)
REG	.059375	4	4	1	1	12	12	15	RV/GV:Z-(EXP)
REG	-1.	5	5	1	3	1	11	16	GV/CV:R+
REG	-1.	6	6	1	3	1	11	17	GV/CV:R-
REG	-1.	5	6	1	1	1	11	18	GV/CV:O+
REG	-1.	5	6	3	3	1	11	19	GV/CV:O-
REG	.190	5	6	1	2	1	1	20	GV/CV:Z+(AIR-IN)
REG	.380	5	6	3	3	1	1	20	GV/CV:Z+(AIR-IN)
REG	.190	5	6	1	2	11	11	21	GV/CV:Z-(AIR-OUT)
REG	.380	5	6	3	3	11	11	21	GV/CV:Z-(AIR-OUT)
REG	-1.	8	8	3	3	6	10	22	IHX.INT:R+
REG	-1.	8	8	3	3	6	10	23	IHX.INT:R-
REG	-1.	8	8	3	3	6	10	24	IHX.INT:O+
REG	-1.	8	8	3	3	6	10	25	IHX.INT:O-
REG	.4869	8	8	3	3	6	6	26	IHX.INT:Z+(NA-IN)
REG	.4869	8	8	3	3	10	10	27	IHX.INT:Z-(NA-OUT)
REG	-1.	2	2	1	3	3	9	28	RV.INTO:SEP.I=1&2:R+
REG	-1.	1	1	1	3	3	9	29	RV.INTO:SEP.I=1&2:R-
REG	-1.	1	1	1	1	4	6	30	RV.EXTO:ASY:O+
REG	-1.	1	1	2	2	4	6	30	RV.INTO:ASY:O+
REG	-1.	1	1	3	3	4	6	30	RV.INTO:ASY:O+
REG	-1.	2	2	1	1	2	9	30	RV.EXTO:PIPE:O+
REG	-1.	2	2	2	2	3	8	30	RV.INTO:PUMP/PIPE:O+
REG	-1.	2	2	3	3	4	9	30	RV.INTO:PUMP/IHX:O+
REG	-1.	1	1	1	1	4	6	31	RV.INTO:ASY:O-
REG	-1.	1	1	2	2	4	6	31	RV.INTO:ASY:O-
REG	-1.	1	1	3	3	4	6	31	RV.EXTO:ASY:O-
REG	-1.	2	2	1	1	3	8	31	RV.INTO:PUMP/PIPE:O-
REG	-1.	2	2	2	2	4	9	31	RV.INTO:PUMP/IHX:O-
REG	-1.	2	2	3	3	7	9	31	RV.EXTO:IHX:O-
REG	-1.	2	2	2	3	3	3	32	RV.INTO:BELOW.PUMP+IHX:Z+
REG	-1.	2	2	1	2	10	10	32	RV.INTO:ABOVE.PIPE+PUMP:Z+
REG	-1.	2	2	2	3	2	2	33	RV.INTO:BELOW.PUMP+IHX:Z-
REG	-1.	2	2	1	2	9	9	33	RV.INTO:ABOVE.PIPE+PUMP:Z-

END
BNDRY. SURF. IDENT.

&DATA

```

IBOIL=0, IFENER=1, IFPROP=0,
ISETEN=1, ISETMO=1, ISTATE=0, NEWTS=1, NEWREB=0, NEWFOR=1,
ITIBUG=2, IENBUG=1, IMOBUG=1,
TSTART=0., NTMAX=1000, TIMAX=1.0E+20, TREST=20.,
IDTIME=2, IHTWAL(99)=2,
DT=0.01,1.0, LASTDT=100, IT=2*1, ITMAXP=249, ITMAXE=249,
EPS1=1.0E-09, EPS2=1.0E-13, EPS3=1.0E-07, EPS5=1.0E-09,
NTHCON=-1,9*0,

```

Table E.1 Listing of Card-Image Input File (Cont'd)

```

KFLOW=-3,1,2*-3,1,-3, -6,1, 2*1,2*-3,2*1,-6, 2*1,2*-3,1,-5,
      4*1,1,-5, 6*1,
VELOC=6*0., 2*0., 7*0., 4*0.,2.484,0.,
      4*0.,1.32,0., 6*0.,
KTEMP=6*400, 400,400, 7*400, 4*400,1,400,
      4*400,1,400, 6*400,
TEMP=4*395.,321.,468., 468.,395., 7*200., 4*70.,38.,100.,
      4*395.,282.,395., 6*395.,
KPRES=6*0, 1,0, 6*0,1, 5*0,1,
      5*0,1, 6*0,
PRES=6*0., 1.013E+5,0., 6*0.,1.013E+5, 5*0.,1.013E+5,
      5*0.,1.013E+5, 6*0.,
KPROP=6*1, 2*1, 7*2, 6*3, 6*1, 6*1,
TVAL=0.,1.E10, 0.,1.E10,
FVAL=1.,1., 1.,1.,
NEND=2, 2, 0,

GRAVX=0., GRAVY=0., GRAVZ=-9.8,
TEMPO(1)=321.1, PRESO(1)=0.1013E+06,
XPRESO(1)=2.2, YPRESO(1)=0., ZPRESO(1)=13.18,
TEMPO(2)=200., PRESO(2)=0.1013E+06,
XPRESO(2)=2.964, YPRESO(2)=0., ZPRESO(2)=13.18,
GASMW(2)=39.9, FCORO(2)=0.,
FCOH(2)=1.4E+05, FC1H(2)=520.0,
FCOMU(2)=2.35E-05, FC1MU(2)=4.33E-08,
FCOK(2)=1.7E-02, FC1K(2)=4.0E-05,
TEMPO(3)=70., PRESO(3)=0.1013E+06,
XPRESO(3)=3.173, YPRESO(3)=0., ZPRESO(3)=12.15,
GASMW(3)=28.8, FCORO(3)=0.,
FCOH(3)=2.8E+05, FC1H(3)=1040.0,
FCOMU(3)=1.81E-05, FC1MU(3)=3.61E-08,
FCOK(3)=2.48E-02, FC1K(3)=5.73E-05,
FCTLO=250., FCTHI=750.,
NMATER=2,
COK= 4.63, 15.0,
C1K= 1.85E-02, 0.013,
C2K= 2.8E-06, 0.0,
C0CP= 210.0, 480.0,
C1CP= 0.0, 0.1736,
C2CP= 0.0, 0.0,
CORO= 15400.0, 7979.0,
C1RO= 0.0, 0.0,
C2RO= 0.0, 0.0,

NTPLOT=-10,0,
NTPRNT=-9999,0,
ISTPR= -06201,06203, -10201,10203, -11201,11203, -12201,12203,
      -01201,01203, -02201,02202, -03201,03203,
      -05201,05203, -08201,08203, -07201,07203,
      -15201,15203,
      -80001,80005, 0,
NTHPR= -01201,01203, -02201,02202, -03201,03203,
      -05201,05203, -08201,08203, -07201,07203,
      -09201,09203, -15201,15203,
      -80001,80005, 0,

NHEATC=7,
HEATC1= 5.0, 5.0, 5.0, 6.16, 1.0, 1.0,
      5.0,

```

Table E.1 Listing of Card-Image Input File (Cont'd)

```

HEATC2= 0.401, 0.283, 0.025, 0.527, 1.22, 1.22,
        0.025,
HEATC3= 0.45, 0.45, 0.8, 0.653, 0.457, 0.457,
        0.8,
HEATC4= 0.45, 0.45, 0.8, 0.653, 0.4, 0.4,
        0.8,

ICORR= 1,2,3, 4,5, 6,6,
FORCEF= 3*0.5, 2*0.5, 2*0.5,
REYLEN= 3*3.39E-03, .0171,.0190, .286,.078,
CLENGTH= 3*3.39E-03, .0171,.0190, .286,.078,
NCORR=6,
ACORRL= 0.,104.7,0., 2*51.9, 64.,
BCORRL= 3*-1., 2*-1., -1.,
CCORRL= .0663,0.,.0037, .0046,.386, 0.,
REYTRN= 3*1674., 2*1189., 1189.,
ACORRT= 0.,.400,0., 2*.256, .316,
BCORRT= 3*-.25, 2*-.25, -.25,
CCORRT= .0663,0.,.0037, .0046,.386, 0.,

IFRAD=2, NRAD=4, OMR=0.05,
EMITS= 4*0.7,
FVIEW= 0., .96543, 0., 0., 95*0.,
        1., .03457, 0., 0., 95*0.,
        0., 0., 0., .90876, 95*0.,
        0., 0., 1., .09124, 95*0.,
        9405*0.,

IFPUMP=4,
TURBV=2.0E-3, TURBC=0.0,
&END
HTEX 2 3 4 1 3 1 11 RV/GV:ARGON
HTEX 2 4 4 1 1 12 12 RV/GV(EXP):ARGON
HTEX 3 5 6 1 3 1 11 GV/CV:AIR
END FLUID CELL LOC.
&PUMP
IPUBUG=3, NPUMP=1, DENRO=8575.,
ICORP(1)=1, QFLOWR(1)=0.64368, PUMPHR(1)=84.7,
TORQR(1)=26981., NTORQM(1)=2,
PUMPS(1)=1116., PUMPSR=1116., PINER(1)=1182.,
QCUT=1.0E-09, CUTPF=0.01,.268,
CORPF=.01,0.,-73.13, .00268,0.,.07, .00383,.01071,.01406,
&END
PUMZ 1 2 2 2 2 7 7 PRI.PUMP
END PUMP LOC.
ZFOR 1 1 1 1 3 3 3 DR: IN
ZFOR 2 1 1 1 3 4 5 DR: PIN
ZFOR 3 1 1 1 3 6 6 DR: OUT
ZFOR 4 2 2 3 3 6 9 IHX(PRI): TUBE
ZFOR 5 8 8 3 3 6 9 IHX(INT): SHELL
ZFOR 6 2 2 1 1 1 8 PIPE
ZFOR 7 2 2 2 2 6 8 PUMP.DUCT
END FORCE STRUCT. LOC.
&STRUCT ITSBUG=0, &END
FUEL PINS (Q>0)
&T N=1, IXYZ=3, NT=1, RODFR=-2495.6, OUTR=3.315E-03, &END
&F IHT=1, HYD=3.39E-03, &END
&M MI=2, NP=1, DR=.457E-03, Q=0., SGAP=.083E-03,HGAP=2.0E+05, &END
&M MI=1, NP=5, DR=.555E-03, Q=539.04E+06, &END

```


Table E.1 Listing of Card-Image Input File (Cont'd)

```

IHX TUBES
&T N=2, IXYZ=3, NT=0, RODFR=1000., OUTR=9.525E-03, &END
&F IHT=4, HYD=19.0E-03, &END
&M MI=2, NP=3, DR=.326E-03, Q=0., &END
&F IHT=3, HYD=17.1E-03, &END
RV WALL
&T N=3, IXYZ=3, NT=0, RODFR=-.7251, OUTR=2.870, &END
&F IHT=5, HYD=0.2032, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=1, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=2, DR=2.539E-2, Q=0., KRAD=0, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=0, &END
&F IHT=7, HYD=0.0762, &END
GV WALL
&T N=4, IXYZ=3, NT=0, RODFR=-.3015, OUTR=3.023, &END
&F IHT=6, HYD=0.610, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=3, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=2, DR=2.539E-2, Q=0., KRAD=0, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=2, &END
&F IHT=5, HYD=0.2032, &END
CV WALL
&T N=5, IXYZ=3, NT=0, RODFR=-.2856, OUTR=3.3528, &END
&M MI=2, NP=1, DR=2.539E-2, Q=0., KRAD=0, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=4, &END
&F IHT=6, HYD=0.610, &END
END THERMAL STRUCT. PROTOTYPE
OUT 1 1 1 1 3 5 5 DR.PIN: FUEL (Q>0)
OUT 2 8 8 3 3 7 9 IHX.INT(SHELL)
IN 2 2 2 3 3 7 9 IHX.PRI(TUBE)
OUT 3 3 3 1 3 1 11 RV:ARGON
IN 3 2 2 1 3 1 11 RV:NA
OUT 4 5 5 1 3 1 11 GV:AIR-FLOW
IN 4 4 4 1 3 1 11 GV:ARGON
IN 5 6 6 1 3 1 11 CV:AIR-FLOW
END THERMAL STRUCT. LOC.
VELB -2.484 5 6 1 3 11 11 21 GV/CV:OUT
VELB -1.32 8 8 3 3 10 10 27 IHX.INT:OUT
END BNDRY. VALUE INIT.
AL .0627 1 1 1 3 4 6 ASY:DR
ALZ .0627 1 1 1 3 3 6 ASY:DR
AL .1022 2 2 3 3 7 9 IHX.PRI:TUBE
ALZ .1022 2 2 3 3 6 9 IHX.PRI:TUBE
AL .0376 2 2 2 2 7 8 PUMP.DUCT
ALZ .0376 2 2 2 2 6 8 PUMP.DUCT
AL .1143 2 2 1 1 3 8 PIPE
ALZ .1143 2 2 1 1 2 8 PIPE
AL .6712 3 4 1 3 1 11 RV/GV:ARGON
AL .6712 4 4 1 1 12 12 RV/GV:ARGON
ALY .6724 3 4 1 2 1 11 RV/GV:ARGON
ALZ .6712 3 4 1 3 1 10 RV/GV:ARGON
ALZ .6712 4 4 1 1 11 11 RV/GV:ARGON
AL .889 5 6 1 3 1 11 GV/CV:AIR(FLOW)
ALY .889 5 6 1 2 1 11 GV/CV:AIR(FLOW)
ALZ .889 5 6 1 3 1 10 GV/CV:AIR(FLOW)
AL .213 8 8 3 3 6 10 IHX.INT:SHELL
ALZ .213 8 8 3 3 6 9 IHX.INT:SHELL
P 1.0E+06 2 2 2 2 8 9 PUMP.OUT
P 1.0E+06 2 2 1 1 3 9 PIPE
P 1.0E+06 2 2 1 3 1 2 PIPE/CORE.IN
P 1.0E+06 1 1 1 3 1 4 CORE.IN

```

Table E.1 Listing of Card-Image Input File (Cont'd)

TL	468.	1	1	1	3	5	11	ASY.EXIT/HOT.POOL
TL	468.	2	2	1	3	10	11	HOT.POOL
TL	468.	2	2	1	1	12	12	HOT.POOL(EXP.CELL)
TL	419.	2	2	3	3	9	9	IHX.PRI
TL	370.	2	2	3	3	8	8	IHX.PRI
TL	395.0	8	8	3	3	6	10	IHX.INT
WL	15.2	2	2	2	2	6	8	PUMP
VL	-.27	2	2	1	1	9	9	PUMP/PIPE
WL	-5.0	2	2	1	1	2	8	PIPE
UL	-0.4	1	1	1	3	1	2	PIPE/CORE.IN
WL	.34	1	1	1	3	2	2	CORE.IN
WL	5.43	1	1	1	3	3	6	ASY:DR
WL	.34	1	1	1	3	7	9	HOT.POOL
UL	.07	1	1	1	3	10	11	HOT.POOL/IHX
WL	-2.85	2	2	3	3	6	9	IHX.PRI:TUBE
WL	-.29	2	2	3	3	3	5	IHX/PUMP
VL	-.97	2	2	2	2	3	3	IHX/PUMP
WL	.57	2	2	2	2	3	5	IHX/PUMP
WL	.001	3	3	1	3	1	10	RV/GV
WL	-.001	4	4	1	3	1	10	RV/GV
WL	2.484	5	6	1	3	1	10	GV/CV
WL	1.32	8	8	3	3	6	9	IHX.INT
END								INTERNAL CELL INIT.

E.2.1 Geometry

This sample problem uses cylindrical (r, θ, z) geometry for the fluid grid, as selected by $IGEOM = -1$ in NAMELIST GEOM. There are 8 nodes in the radial direction ($IMAX = 8$ in NAMELIST GEOM), a choice which is motivated by the location of the four different fluid types: nodes 1 and 2 are the sodium within the RV, nodes 3 and 4 are the argon between the RV and the GV, nodes 5 and 6 are the air between the GV and the CV, node 7 contains no fluid cells (but provides a radial spacing offset in the grid), and node 8 is the sodium in the intermediate side of the IHX; the thicknesses of these nodes are input using array DX in NAMELIST GEOM. There are 3 nodes in the azimuthal direction ($JMAX = 3$), a choice motivated by the need to define different flow regions in radial node 2 within the RV: node 1 contains a pipe, node 2 contains a pump, and node 3 contains the IHX; the thicknesses of these nodes are input using array DY in NAMELIST GEOM. There are 12 nodes in the axial direction ($KMAX = 12$ in NAMELIST GEOM), a choice motivated by the need to define different flow geometries within the RV: nodes 1 and 2 are the cold pool, the top of node 3 defines the bottom of the fuel assemblies, nodes 4 through 6 are the fuel assemblies, nodes 7 through 9 are the IHX, nodes 10 and 11 are the hot pool, and node 12 is used for an expansion cell; the thicknesses of these nodes are input using array DZ in NAMELIST GEOM. (The user is cautioned that this sample problem has a rather coarse noding and will probably result in a rather inaccurate representation of the physical system; a more realistic representation of this physical system would normally use 2 to 3 (or more) times the number of fluid cells used here in each of the three coordinate directions.)

Several views of the grid system for this problem are shown in Figs. E.1 through E.3; these figures were produced using the VECT program by specifying $IARROW = 0$ and $IGRID = 1$ (cf. Sec. D.2). In all three figures, solid lines are used to denote boundary surfaces (as defined in the second paragraph below), whether impervious or pervious; solid lines are also used to denote impervious cell faces which are not on boundary surfaces; dotted lines are used to denote cell faces which are pervious but not on boundary surfaces. Figure E.1 shows an elevation view having J plane number 1 on the right and J plane number 3 on the left; the centerline of the RV is represented by the two vertical dotted lines immediately adjacent to each other near the center

of the page; the radial cell location index, I , is 1 for the vertical column of cells on each side of this line, and the I index increases as one moves to the left or right from the center; the axial cell location index, K , is 1 for the horizontal row of cells at the bottom, and the K index increases as one moves from the bottom to the top. The solid lines within the RV denote the shaping of the flow path for the IHX ($J=3$) and for the pipe ($J=1$); the two cells extending above the top of the other cells (at positions (I,J,K) of $(2,1,12)$ and $(4,1,12)$) are the two expansion cells; the five cells on the left side of the figure at locations $(8,3,6)$ through $(8,3,10)$ are the intermediate side of the IHX. Figure E.2 shows an unrolled elevation view of I plane number 2; the azimuthal cell location index, J , is 1 for the column of cells at the left side, and the J index increases as one moves from left to right; the axial cell location index, K , is 1 for the row of cells at the bottom, and the K index increases as one moves from the bottom to the top. This figure shows further details of the shaping of the flow path which will ultimately allow for flow down through the primary side of the IHX (cells having $J=3$), up through the pump (cells having $J=2$), and down through the pipe (cells having $J=1$). Figure E.3 shows a plan view of K plane number 9; the radial cell index, I , is 1 for the ring of cells closest to the bottom left-hand corner (which is the centerline of the RV), and the I index increases as one moves outward from this central ring; the azimuthal cell index, J , is 1 for the line of cells having a face on the horizontal line at the bottom of the figure, and the J index increases as one moves counterclockwise from this position. The one large cell at $I=8$ and $J=3$ in Fig. E.3 is part of the intermediate side of the IHX.

$J = 3, 1$

Fig. E.1 Elevation View of Geometry Grid for Sample Problem
for Planes $J=3$ (left) and $J=1$ (right).

$I = 2$

Fig. E.2 Elevation View of Geometry Grid for Sample Problem for Plane $I=2$.

$K = 9$

Fig. E.3 Plan View of Geometry Grid for Sample Problem for Plane K=9.

The cylinder is oriented with its positive z-axis pointing upward, as indicated by the input of gravitational acceleration vector components $\text{GRAVX}=\text{GRAVY}=0.0$ and $\text{GRAVZ}=-9.8 \text{ m/s}^2$ in NAMELIST DATA.

The multiple combinations of orientation, fluid type, and boundary condition type dictate that 33 boundary surfaces are needed ($\text{NSURF}=33$ in NAMELIST GEOM). These boundary surfaces are summarized in Table E.2. Boundary surfaces 1 through 8 and 28 through 33 define the space within the RV, surfaces 9 through 15 define the space between the RV and the GV, surfaces 16 through 21 define the space between the GV and the CV, and surfaces 22 through 27 define the intermediate side of the IHX. The components of the unit surface-normal vector are input using arrays XNORML , YNORML , and ZNORML in NAMELIST GEOM.

Table E.2 Boundary Surface Summary

Surface Number	Surface Normal Direction	KFLOW (VELOC)	KPRES (PRES)	KTEMP (TEMP)	KPROP	Other
1	+r	-3	0	400	1	RV
2	-r	1(0.)	0	400	1	RV
3	+ θ	-3	0	400	1	RV
4	- θ	-3	0	400	1	RV
5	+z	1(0.)	0	400	1	RV
6	-z	-3	0	400	1	RV
7	-z	-6	1(1.013E5)	400	1	RV:exp.
8	-r	1(0.)	0	400	1	RV
9	+r	1(0.)	0	400	2	RV/GV
10	-r	1(0.)	0	400	2	RV/GV
11	+ θ	-3	0	400	2	RV/GV
12	- θ	-3	0	400	2	RV/GV
13	+z	1(0.)	0	400	2	RV/GV
14	-z	1(0.)	0	400	2	RV/GV
15	-z	-6	1(1.013E5)	400	2	RV/GV:exp.
16	+r	1(0.)	0	400	3	GV/CV
17	-r	1(0.)	0	400	3	GV/CV
18	+ θ	-3	0	400	3	GV/CV
19	- θ	-3	0	400	3	GV/CV
20	+z	1(2.484)	0	1(38.)	3	GV/CV:in
21	-z	-5	1(1.013E5)	400	3	GV/CV:out
22	+r	1(0.)	0	400	1	IHX _{in}
23	-r	1(0.)	0	400	1	IHX _{in}
24	+ θ	1(0.)	0	400	1	IHX _{in}
25	- θ	1(0.)	0	400	1	IHX _{in}
26	+z	1(1.32)	0	1(282.)	1	IHX _{in} :in
27	-z	-5	1(1.013E5)	400	1	IHX _{in} :out
28	+r	1(0.)	0	400	1	RV
29	-r	1(0.)	0	400	1	RV
30	+ θ	1(0.)	0	400	1	RV
31	- θ	1(0.)	0	400	1	RV
32	+z	1(0.)	0	400	1	RV
33	-z	1(0.)	0	400	1	RV

The Boundary Surface Identification input block (the records beginning with "REG " following NAMELIST GEOM) include one or more records for each boundary surface to specify which fluid cells are adjacent to each boundary surface. For example, the first such record indicates that fluid cells having spatial location indices $I=1$, $1 \leq J \leq 3$, and $1 \leq K \leq 11$ are on the +I side ($XNORML(1)=1.0$, $YNORML(1)=ZNORML(1)=0.0$) of boundary surface number 1; this is the centerline of the RV. The comment portion of each record (i.e. positions 43 through 80) have been used to indicate the physical meaning of the boundary surface. Most records have specified the AREA of the boundary surface element as a negative quantity, which indicates the area is to be computed internally by the computer program based on the grid dimensions; in a few instances (e.g. the inlets and outlets of the GV/CV gap and the IHX), the actual area has been specified (indicated by a positive value), since the flow area differs from the value which would be computed using the grid dimensions; the fluid cells adjacent to these boundary surfaces will be assigned nonunity volume porosities, as described in the third paragraph below. Note that pairs of coincident boundary surfaces, having unit surface-normal vectors pointing in opposite directions and a fixed velocity of zero, are used to separate fluid regions; examples: (1) boundary surface numbers 2 and 9 separate the sodium in the RV from the argon in the RV/GV gap; (2) boundary surface numbers 10 and 16 separate the argon in the RV/GV gap from the air in the GV/CV gap; (3) most boundary surface elements defined for boundary surface numbers 28, 30, and 32 are paired with boundary surface elements defined for boundary surface numbers 29, 31, and 33, respectively, to define impervious surfaces within the RV. The record beginning with "END " terminates the Boundary Surface Identification input block.

The number of fluid cells (=205) and the number of boundary surface elements (=482) are input using variables NM1 and NL1, respectively, in NAMELIST GEOM. The appropriate input values are computed by counting the cells and cell faces, respectively, defined on the records in the Boundary Surface Identification input block.

The use of three fluids is indicated by IFHTX=3 in NAMELIST GEOM. By default, fluid number 1 (sodium) is placed in all cells. The records in the Fluid Cell Location input block (records beginning with "HTEX" between NAMELISTs DATA and PUMP) specify that

this default is overridden as follows: (1) fluid number 2 (argon) is to be placed in the cells between the RV and the GV and (2) fluid number 3 (air) is to be placed in the cells between the GV and the CV. An "END " record follows the last "HTEX" record to terminate the Fluid Cell Location input block. The specification of which fluid number is adjacent to each boundary surface is accomplished using the KPROP array in NAMELIST DATA. (The input of fluid properties is covered in Sec. E.2.7.)

By default, all fluid cells have volumes and cell-face areas based on the grid spacings input in arrays DX, DY, and DZ -- i.e. the default volume and surface porosities are 1. To override these defaults, nonunity volume and surface porosities may be input in the Internal Cell Initialization input block (which is the last block in the card-image input file) using records which begin with "AL ", "ALX ", "ALY ", and "ALZ ". As shown in Table E.1, this has been done for the fuel assemblies, the IHX, the pump, the pipe, the RV/GV gap, and the GV/CV gap. As an example of the values to input, consider the pump; the physical pump to be modeled has a uniform flow area of 0.04223 m^2 ; the pump is to be placed in cells having spatial location indices $I=2$, $J=2$ and $7 \leq K \leq 8$, for which the axial flow area based on the grid dimensions is $\{DX(2)*[DX(1)+0.5*DX(2)]*DY(2)=\} 1.1231 \text{ m}^2$; therefore, the volume and z-direction surface porosities are input as the ratio of physical to grid flow areas, or 0.0376. Note that the range of cell indices specified for surface porosities is usually one bigger than the range of cell indices specified for volume porosities for each region; this is due to the fact that volume porosities are input for fluid cells and surface porosities are input for faces of fluid cells. Using the pump as an example, the volume porosities are input for fluid cells having spatial indices $7 \leq K \leq 8$; however, in order to have the axial flow area constant throughout the entire range of these cells, the z-direction surface porosity must be input for spatial indices $6 \leq K \leq 8$. A similar procedure is followed to define the space between the RV and the GV and between the GV and the CV, since the locations of the radial grid lines (input using array DX) do not correspond exactly to either the inner or outer surfaces of these vessels.

E.2.2 Boundary and Initial Conditions

Three types of fluid boundary conditions must be specified for each boundary surface: velocity, pressure, and energy. This information is input using arrays in NAMELIST DATA.

The type of velocity boundary condition is indicated using array KFLOW. Most boundary surfaces are zero slip, indicated by $KFLOW(n)=1$ and $VELOC(n)=0.0$. The free-slip boundary condition ($KFLOW(n)=-3$) is specified for boundary surfaces corresponding to the centerline of the RV ($n=1$), the symmetry cuts on the $\pm\theta$ faces ($n=3, 4, 11, 12, 18$, and 19), and the top surface of the RV ($n=6$, to recognize that the sodium in the RV is not actually in contact with the RV cover). The velocity is specified as constant ($KFLOW(n)=1$ and $VELOC(n)>0$) for the inlet to the GV/CV gap ($n=20$) and the intermediate side of the IHX ($n=26$); a continuative mass flow boundary condition ($KFLOW(n)=-5$) is specified for the exits of these two regions ($n=21$ and 27 , respectively). An expansion cell is provided for the RV interior and for the RV/GV gap, which have no bulk inlet or outlet flow; one face of each of these cells has an expansion-cell velocity boundary condition specified ($KFLOW(n)=-6$ for $n=7$ and $n=15$).

The type of pressure boundary condition is specified using array KPRES. The pressure is specified as atmospheric for both expansion cells, the exit of the GV/CV gap, and the exit of the intermediate side of the IHX ($KPRES(n)=1$ and $PRES(1)=1.013E5$ for $n=7, 15, 21$, and 27). All other boundary surfaces must use the not-specified pressure boundary condition ($KPRES=0$).

The type of energy boundary condition is specified using array KTEMP. The temperature is specified as constant at the inlets to the GV/CV gap ($n=20$) and the intermediate side of the IHX ($n=26$) by having $KTEMP(n)=1$; the actual temperature value for each of these two boundary surfaces is input using the corresponding element in array TEMP. All other boundary surfaces are adiabatic ($KTEMP(n)=400$). (Energy transfer among regions will be accomplished using Thermal Structures (cf. Sec. E.2.3) rather than using the available fluid energy boundary conditions.)

Initial conditions are specified for each fluid number ($1 \leq n \leq 3$) using arrays in NAMELIST DATA. A uniform temperature is assigned for each fluid using array TEMP0; for fluid number 1, the value input is the expected cold pool temperature in the RV; the values input for the other two regions are expected average values. A reference pressure (using array PRES0) and corresponding Cartesian-coordinate position (using arrays XPRES0, YPRES0, and ZPRES0) are input for each fluid; these pieces of information, along with the fluid density at the initial temperature (from array TEMP0) and the components of the gravitational acceleration vector (from GRAVX, GRAVY, and GRAVZ), are used to establish an initial pressure distribution based on the gravity head. (Since $\text{GRAVX} = \text{GRAVY} = 0$, the values input for arrays XPRES0 and YPRES0 are irrelevant in this sample problem.) These initializations are overridden for some cells by records specified in the Internal Cell Initialization input block at the end of the card-image input file. The records beginning with "P " are used to initialize a higher pressure in the RV cells between the pump exit and the fuel assembly inlet. The records beginning with "TL " are used to initialize higher temperatures in the RV along the flow path between the fuel assembly exit and the IHX midplane and on the intermediate side of the IHX. The records beginning with "UL ", "VL ", and "WL " are used to initialize a velocity distribution within each region. The record beginning with "END " following the velocity initialization records denotes the end of the Internal Cell Initialization input block (and, thus, the end of the card-image input file). Since this is the beginning of a steady-state calculation, these initial values are merely guesses used to get the solution started; specification of good guesses will shorten the time required by the computer program to obtain a converged solution.

Records ("VELB ...") are present in the Boundary Value Initialization input block in order to initialize velocities for two exit-type boundary surfaces: the GV/CV gap (boundary surface number 21) and the intermediate side of the IHX (boundary surface number 27). Note that the specified velocity value is negative in both records, since the direction of flow (upward) is opposite to the direction of the unit surface-normal vector (downward). This input block is near the end of the card-image input file, immediately preceding the Internal Cell Initialization input block. These two records are followed by an "END " record to mark the end of the

Boundary Value Initialization input block. (As was the case for velocities, temperatures, and pressures in the Internal Cell Initialization input block, these two boundary surface velocity initializations are merely guesses and could have been omitted.)

Information has been supplied to define two transient functions using arrays TVAL, FVAL, and NEND in NAMELIST DATA. Both functions are defined to have the constant value of 1.0 for all time. These definitions are not essential for a steady-state calculation; they have been included in the input at this point as a reminder that two transient functions are referenced in other sections of the input (thermal structure and pump); these two functions must have meaningful definitions if the results of the steady-state calculation are later used to begin a transient calculation (cf. Sec. E.5). Note that a 0 is entered as the third element of array NEND to insure proper processing of the preceding entries.

E.2.3 Thermal Structures

The presence of input for the thermal structure model is denoted by the combination of ISTRUC=2 in NAMELIST GEOM and NEWTS=1 in NAMELIST DATA. The remainder of the thermal structure input consists of (in order) NAMELIST STRUCT, the Thermal Structure Prototype input block, and the Thermal Structure Location input block; these three sets of input immediately precede the Boundary Value Initialization input block.

The single variable input in NAMELIST STRUCT (ITSBUG=0) indicates that a debug version of the Thermal Structure Control Block is not to be printed.

There are five thermal structure prototypes defined: (1) fuel pin, (2) IHX tube, (3) RV wall, (4) GV wall, and (5) CV wall. Each prototype has input in both the Thermal Structure Prototype and Thermal Structure Location input blocks. The input is discussed below by prototype. The record preceding each NAMELIST T in the Thermal Structure Prototype input block is a comment, denoted by the first four character positions being blank.

The definition of the prototype for the fuel pins begins with the first NAMELIST T following NAMELIST STRUCT. This is prototype number 1 ($N=1$). The geometry is a cylinder whose axis is parallel to the z axis of the fluid grid ($IXYZ=3$). Since this is a cylinder, an outer radius (OUTR) is specified. The negative value of RODFR indicates that this basic cylinder is to be repeated the indicated number of times per square meter of z -direction grid area (defined as $\Delta r_i \cdot r_{cr,i} \cdot \Delta \theta_j$) for each fluid cell to which the prototype is associated. The specification of $NT=1$ is to allow the internal heat source in the fuel to be multiplied by transient function number 1 in a later transient calculation run. Since a NAMELIST F immediately follows the NAMELIST T, fluid cells will be in thermal contact with the outside of the fuel pins; the heat transfer coefficient to be used between the outer surface of the fuel pins and the associated fluid cells will be calculated using correlation number 1 ($=IHT$) and the characteristic length (HYD) to be used in this heat transfer calculation is the fuel assembly hydraulic diameter (3.39 mm). The NAMELIST F is followed by two NAMELISTs M, denoting two material regions in this prototype. The first (or outer) material region is the cladding; it is composed of material number ($MI=$) 2, has 1 node ($=NP$) of thickness ($DR=$) 0.457 mm, and no heat source ($Q=0$). This material region is separated from the inner material region by a gap of thickness ($SGAP=$) 83 μm , and the gap heat transfer coefficient ($SGAP$) is 0.2 $\text{MW/m}^2 \cdot ^\circ\text{C}$. The second (or inner) material region is the fuel; it is composed of material number ($MI=$) 1, has 5 nodes ($=NP$) of thickness ($DR=$) 0.555 mm, and a heat source of ($Q=$) 539.04 MW/m^3 . Since this is the innermost material region, values are not specified for $SGAP$ and $HGAP$. The absence of a NAMELIST F following the two NAMELISTs M indicates that fluid cells do not interact with the inner surface of this prototype. The first record ("OUT 1 ...") in the Thermal Structure Location input block shows that the outer surface of thermal structure prototype number 1 interacts with fluid cells having spatial location indices $I=1$, $1 \leq J \leq 3$, and $K=5$. The product of the z -direction grid area (1.887 m^2) and RODFR shows that 4709 fuel pins are being represented in this 90° -sector model. (The input for material properties and heat transfer correlations are discussed in Sec. E.2.7.)

The definition of the prototype for the IHX tubes begins with the second NAMELIST T in the Thermal Structure Prototype input block. This is prototype number ($N=$) 2. The geometry is a cylinder whose axis is parallel to the z axis of the fluid grid ($IXYZ=3$). Since

this is a cylinder, an outer radius (OUTR) is specified. The positive value of RODFR indicates that this basic cylinder is to be repeated the indicated number of times per fluid cell (not per cell area) for each fluid cell to which the prototype is associated. The specification of NT=0 indicates that the internal heat source in this prototype is not a function of time during a transient calculation; even though this prototype contains no heat source, NT=0 should still be specified. Since a NAMELIST F immediately follows the NAMELIST T, fluid cells will be in thermal contact with the outside of the tubes; the heat transfer coefficient to be used between the outer surface of the tubes and the associated fluid cells will be calculated using correlation number 4 (=IHT) and the characteristic length (HYD) to be used in this heat transfer calculation is the outer diameter of the tube. The NAMELIST F is followed by one NAMELIST M, denoting one material region in this prototype. This material region is composed of material number (MI=) 2, has 3 nodes (=NP) of thickness (DR=) 0.326 mm, and no heat source (Q=0). Since a NAMELIST F immediately follows the NAMELIST M, fluid cells will also be in thermal contact with the inside of the tubes; the heat transfer coefficient to be used between the inner surface of the tubes and the associated fluid cells will be calculated using correlation number 3 (=IHT) and the characteristic length (HYD) to be used in this heat transfer calculation is the inner diameter of the tube. The second and third records in the Thermal Structure Location input block identify the fluid cells associated with the surfaces of this prototype: the outside surface of the tubes interacts with fluid cells having spatial location indices I=8, J=3, and $7 \leq K \leq 9$ (i.e. the intermediate loop sodium is on the shell side of the IHX) and the inside surface of the tubes interacts with fluid cells having spatial location indices I=2, J=3, and $7 \leq K \leq 9$ (i.e. the primary sodium in the RV is on the tube side of the IHX); therefore, this thermal structure thermally couples cell (2,3,7) to cell (8,3,7), cell (2,3,8) to cell (8,3,8), and cell (2,3,9) to cell (8,3,9) and, thus, provides one of the two paths by which heat is removed from the RV.

Since the remaining three prototypes are also cylinders, the discussion of them will only highlight the aspects of the input which differ in concept from the first two prototypes.

The RV wall is represented by prototype number 3. The negative RODFR value results in there being one quarter of a cylinder in the 90°-sector being modeled in this sample problem.

Fluid cells interact with both the inner and outer surfaces. The characteristic length for heat transfer (HYD) for the outer surface is the hydraulic diameter for the RV/GV gap. The thickness of this wall has been divided into three material regions which differ from each other in two respects: (1) the innermost and outermost partitions are very thin, so that their temperatures are representative of actual surface temperatures; (2) the outer surface is defined as radiation surface number 1 (=KRAD). (Further discussion of radiation heat transfer follows the discussion of the five thermal structures.) Since there is no physical gap between these three material regions, a small SGAP and a large HGAP are specified. The fourth and fifth records in the Thermal Structure Location input block show that this thermal structure thermally couples fluid cells at $I=2$ (sodium in the RV) to fluid cells at $I=3$ (argon in the RV/GV gap) for $1 \leq J \leq 3$ and $1 \leq K \leq 11$ and is, thus, the second path for removing heat from the RV sodium.

The GV wall is represented by prototype number 4. The input for this prototype is very similar in concept to that specified for prototype number 3. The inner surface of this prototype is radiation surface number 2 and the outer surface is radiation surface number 3. The sixth and seventh records in the Thermal Structure Location input block show that this thermal structure thermally couples fluid cells at $I=4$ (argon in the RV/GV gap) to fluid cells at $I=5$ (air in the GV/CV gap) for $1 \leq J \leq 3$ and $1 \leq K \leq 11$.

The CV wall is represented by prototype number 5. The input for this prototype is very similar in concept to that specified for prototype numbers 3 and 4. Since there is no NAMELIST F immediately following the NAMELIST T, the outer surface of this prototype does not interact with fluid cells (and is, thus, adiabatic). Fluid cells only interact with the inner surface, which is also defined as radiation surface number 4. The eighth record in the Thermal Structure Location input block shows that this thermal structure is coupled to fluid cells at $I=6$, $1 \leq J \leq 3$, and $1 \leq K \leq 11$ (air in the GV/CV gap).

An "END " record is present as the last record in both the Thermal Structure Prototype and Location input blocks as a terminator.

Additional input is specified in NAMELIST DATA for the radiation heat transfer model. The use of radiation surfaces is indicated by IFRAD=2, and the number of surfaces (=4) is specified using input variable NRAD. The four radiation surfaces were defined by specifying KRAD>0 in the Thermal Structure Prototype input block: number 1 is the outer surface of the RV wall, number 2 is the inner surface of the GV wall, number 3 is the outer surface of the GV wall, and number 4 is the inner surface of the CV wall. The emissivity for each surface is input using array EMITS. Element (n,m) in array FVIEW is the fraction of radiation leaving surface number n which arrives at surface number m. The values input have been computed under the assumption that the three cylinders which represent the RV, GV, and CV walls are concentric and infinitely long. Considering the radiation surfaces in pairs (number 1 paired with number 2 and number 3 paired with number 4) under these assumptions, there are three bases for entries in the FVIEW array. (1) All of the radiation leaving the outer surface of an inner cylinder reaches the inner surface of an outer cylinder; this defines $FVIEW(1,2)=FVIEW(3,4)=1$, $FVIEW(1,m)=0$ for $m \neq 2$, and $FVIEW(3,m)=0$ for $m \neq 4$. (2) The fraction of radiation leaving the inner surface of an outer cylinder which reaches the outer surface of an inner cylinder is computed using the value from item 1 and the reciprocity law (cf. Eq. 5.2); this defines $FVIEW(2,1)$ and $FVIEW(4,3)$ as nonzero, $FVIEW(2,m)=0$ for $m \geq 3$, and $FVIEW(4,m)=0$ for $m \leq 2$. (3) The fraction of radiation leaving the inner surface of an outer cylinder which reaches other parts of the inner surface of this outer cylinder (i.e. self view) is one minus the fraction of the radiation leaving the inner surface of an outer cylinder which reaches the outer surface of an inner cylinder; this defines $FVIEW(2,2)=1-FVIEW(2,1)$ and $FVIEW(4,4)=1-FVIEW(4,3)$. Zeros are entered as place fillers where needed in specifying the input for this PNRAD*PNRAD two-dimensional array (e.g. $FVIEW(n,m)=0$ for (a) $1 \leq n \leq 4$ and $5 \leq m \leq PNRAD$ and (b) $5 \leq n \leq PNRAD$ and $1 \leq m \leq PNRAD$). (As discussed in Sec. C.1.1, the value of PNRAD is set to 99 by a PARAMETER statement in the source code.) A small value is entered for the radiation heat transfer underrelaxation factor (OMR) since this is the beginning of a steady-state calculation.

E.2.4 Force Structures

The use of the force structure model is indicated by $NFORCE > 0$ in NAMELIST GEOM. The value entered for $NFORCE$ ($=7$) is the number of force structure prototypes. The presence of the force structure model input in the card-image input file is indicated by $NEWFOR=1$ in NAMELIST DATA; the input consists of arrays in NAMELIST DATA and records in the Force Structure Location input block (which is just before NAMELIST STRUCT). The seven force structure prototypes invoke six friction factor correlations, as specified by $NCORR$ in NAMELIST DATA. The remainder of the input is discussed below by prototype.

The first three force structure prototypes are used for the fuel assembly region. As seen in the first three records of the Force Structure Location input block, these are forces applied to z-direction momentum cells having location indices $I=1$ and $1 \leq J \leq 3$. The second prototype is applied to the momentum cells between (but excluding) the inlet and the exit cells (i.e. $4 \leq K \leq 5$); the nominal value is specified for the overall force multiplier ($FORCEF(2)=0.5$), and both characteristic lengths ($REYLEN(2)$ and $CLENTN(2)$) are input as the hydraulic diameter (3.39 mm) of the fuel assembly; the friction factor (f) is computed using correlation number 2 ($=ICORR(2)$). Friction factor correlation number 2 is a function of Reynolds number (Re) given by $f=104.7/Re$ in the laminar range and $f=0.400/Re^{0.25}$ in the turbulent range (as specified by $xCORRy(2)$, for $x=A,B,C$ and $y=L,T$); the coefficients specified are specifically for this bundle, rather than being generic values; the transition Reynolds number ($REYTRN(2)$) is input as the value at which these two friction factor correlations actually intersect. The first and third force structure prototypes are applied to the momentum cells at the inlet ($K=3$) and outlet ($K=6$), respectively; the inlet and outlet friction factor correlations (selected by $ICORR(1)=1$ and $ICORR(3)=3$) are independent of the Reynolds number; therefore, $ACORRL(n)=ACORRT(n)=0$ and $CCORRL(n)=CCORRT(n)>0$ are input for $n=1$ and $n=3$.

Force structure prototype numbers 4 and 5 are used for the IHX. As seen in the fourth and fifth records of the Force Structure Location input block, these are forces applied to z-direction momentum cells having location indices $J=3$ and $6 \leq K \leq 9$. The fourth prototype is applied to the momentum cells on the primary side of the IHX (i.e. $I=2$); the nominal value is

specified for the overall force multiplier ($\text{FORCEF}(4)=0.5$), and both characteristic lengths ($\text{REYLEN}(4)$ and $\text{CLENTH}(4)$) are input as the inner diameter (17.1 mm) of the IHX tubes; the friction factor (f) is computed using correlation number 4 ($=\text{ICORR}(4)$). Friction factor correlation number 4 is a constant plus a function of Reynolds number (Re) given by $f=0.0046+51.9/\text{Re}$ in the laminar range and $f=0.0046+0.256/\text{Re}^{0.25}$ in the turbulent range (as specified by $\text{xCORRy}(4)$, for $x=A,B,C$ and $y=L,T$); the coefficients have been adjusted downward from nominal values by the ratio of the real length of the IHX ($L_{\text{IHX}} = \text{DZ}(7)+\text{DZ}(8)+\text{DZ}(9) = 5.34$ m) to the length of the z-direction momentum cells to which the force is applied ($L_{\text{IHX}}+0.5*(\text{DZ}(6)+\text{DZ}(10)) = 6.58$ m); the transition Reynolds number ($\text{REYTRN}(4)$) is input as the value at which these two friction factor correlations actually intersect. The fifth prototype is applied to the momentum cells on the intermediate side of the IHX (i.e. $I=8$); the input is similar to that used for the fourth prototype except the characteristic length is the outer diameter of the tube and the friction factor is obtained from correlation number 5. Friction factor correlation numbers 4 and 5 differ only in the value of the constant part ($\text{CCORRy}(n)$ for $y=L,T$).

Force structure prototype number 6 is used for the pipe, which runs from the pump exit to the cold pool. As seen in the sixth record of the Force Structure Location input block, these are forces applied to z-direction momentum cells having location indices $I=2$, $J=1$, and $1 \leq K \leq 8$. The nominal value is specified for the overall force multiplier ($\text{FORCEF}(6)=0.5$), and both characteristic lengths ($\text{REYLEN}(6)$ and $\text{CLENTH}(6)$) are input as the inner diameter (0.286 m) of the pipe; the friction factor is computed using correlation number 6 ($=\text{ICORR}(6)$). Friction factor correlation number 6 is the familiar function of Reynolds number given by $f=64/\text{Re}$ in the laminar range and $f=0.316/\text{Re}^{0.25}$ in the turbulent range (as specified by $\text{xCORRy}(6)$, for $x=A,B,C$ and $y=L,T$); the transition Reynolds number ($\text{REYTRN}(6)$) is input as the value at which these two friction factor correlations actually intersect.

Force structure prototype number 7 is used for the pump. As seen in the seventh record of the Force Structure Location input block, these are forces applied to z-direction momentum cells having location indices $I=2$, $J=2$, and $6 \leq K \leq 8$. The nominal value is specified for the overall force multiplier ($\text{FORCEF}(7)=0.5$), and both characteristic lengths ($\text{REYLEN}(7)$ and

CLENTH(7)) are input as the effective hydraulic diameter (0.078 m) of the pump; the friction factor is computed using correlation number 6 (=ICORR(6)), which is described in the preceding paragraph.

An "END " record is present at the at the end of the Force Structure Location input block as a terminator.

E.2.5 Pump

The use of the pump model is indicated by IFPUMP>0 in NAMELIST DATA. The value entered for IFPUMP (=4) selects the "homologous with reverse flow precluded" model to be used for all pumps in the calculation. The rest of the pump input consists of NAMELIST PUMP and the Pump Location input block, which are located after the Fluid Cell Location input block.

The variables input in NAMELIST PUMP are as follows. There is one pump in the calculation (NPUMP=1). This pump will use the general homologous curves (ICORP(1)=1). The reference pump conditions are a volumetric flow rate (QFLOWR(1)) of 0.64368 m³/s, a head (PUMPHR(1)) of 84.7 m (0.726 Mpa), speed (PUMPSR(1)) of 1116 rev/min, torque (TORQR) of 0.026981 MN·m, product of density and gravitational acceleration (DENRO) of 8575 kg/m²·s². The default frictional torque function (WF($\tilde{N}$)) has been overridden by specifying input in arrays CORPF and CUTPF. The actual pump speed (PUMPS(1)) has been specified to be the same as the reference speed; the actual pump head, flow rate, and torque are a result of the steady-state calculation. Note that speed and head are not input in SI Units. (Values have also been specified for the moment of inertia (PINER(1)=1182 kg·m²), a transient function number for motor torque (NTORQM(1)=2), and essentially no use of the linear head versus flow relationship (QCUT $\approx$ 0); however, none of these values will be used until a transient calculation is begun.)

The first record in the Pump Location input block specifies that the pressure increase supplied by the pump will be applied in the z direction between K=7 and K=8 at I=2 and J=2.

The "END " record is present at the end of this block as a terminator. Records are present in the Boundary Surface Identification input block for boundary surface numbers 28 through 31 to preclude flow in the r and θ directions in the vicinity of the pump duct.

E.2.6 Turbulence

The "user-specified constant-in-space" turbulence model is selected by ITURKE=0 in NAMELIST GEOM. In NAMELIST DATA, the turbulent viscosity (TURBV) is input as 0.002 Pa·s and the turbulent thermal conductivity (TURBC) is input as 0.

E.2.7 Properties and Heat Transfer Coefficients

As noted in Sec. E.2.1, there are three fluids in the calculation: (1) sodium, (2) argon, and (3) air. Fluid number 1 will use "detailed" properties, as denoted by IFPROP=0 in NAMELIST DATA; no other input is required for fluid number 1 in the card-image input file; however, the user must link to the sodium properties when forming an executable load module to execute this sample problem (cf. Secs. 9.1, B.1, and C.2). The other two fluids must use "simplified" properties; this input is provided in NAMELIST DATA using elements 2 and 3 of arrays FCiH, FCiMU, FCiK, and FCiRO or GASMW, for $i=0$ and $i=1$, for enthalpy, viscosity, thermal conductivity, and density, respectively; since both of these fluids are gases, the ideal-gas-law density form is selected by specifying GASWM>0 and FCORO=0. Additional input denotes that tables summarizing the fluid property information will be printed for the temperature range (FCTLO=) 250 °C to (FCTHI=) 750 °C for all fluids.

Specification of NMATER=2 in NAMELIST DATA indicates that properties will be provided for two structure materials; referring back to the thermal structure input (MI values in NAMELISTs M), these materials are as follows: (1) a metal-based (U-Pu-Zr) fuel and (2) steel (i.e. fuel pin cladding, IHX tubes, all vessel walls). The actual properties are provided using elements 1 and 2 of arrays CiK, CiCP, and CiRO, for $i=0$ through 2, in NAMELIST DATA for the thermal conductivity, specific heat capacity, and density, respectively.

Specification of $NHEATC=7$ in NAMELIST DATA indicates that seven fluid/structure heat transfer coefficients will be provided; referring back to the thermal structure input (IHT values in NAMELISTs F), these correlations are used as follows: (1) fuel-pin/sodium, (2) not used (reserved for blanket assemblies), (3) IHX tube side, (4) IHX shell side, (5) RV/GV gap, (6) GV/CV gap, and (7) inside RV wall. Since all of these correlation numbers are less than or equal to 20, they will all be computed using the general form (Eq. 9.11). The actual constants used in the general form are provided using elements 1 through 7 of arrays $HEATCi$, for $i=1$ through 4, in NAMELIST DATA. (These correlations should not be considered generic.)

E.2.8 Auxiliary

The hexagonal fuel assembly model is not used (since $IGEOM=-1$ in NAMELIST GEOM). The combination of $IFREB=0$ in NAMELIST GEOM and $NEWREB=0$ in NAMELIST DATA indicates that the mass rebalancing model is not used.

The input values associated with starting, stopping, and restarting the case are described in this paragraph. Specifying $IFRES=1$ in NAMELIST GEOM indicates that a restart file is written at the end of this steady-state run. The other input variables are in NAMELIST DATA. The input choice of $ISTATE=0$ specifies that this run is the beginning of a steady-state calculation; correspondingly, the initial case time (TSTART) is 0. Large values are input for the ending case time (TIMAX) and time step number (NTMAX), so that the calculation will continue (within the CPU time limit) until steady state is actually reached. When the CPU time remaining for this run drops below 20 s ($=TREST$), this run will be terminated; this time is reserved for writing the restart file ($IFRES=1$) and final long edit ($NTPRNT=-9999$ below). The input of $IBOIL=0$ specifies that no check will be performed for boiling of fluid number 1 (sodium).

The input values associated with solution methods are discussed in this paragraph. The pressure equation system will be solved using the preconditioned conjugate gradient method ($ISYMCH=5$ in NAMELIST GEOM), and the fluid energy equation system will be solved using

the successive overrelaxation method (IFITEN=3 in NAMELIST GEOM). The remaining input associated with solution methods is in NAMELIST DATA. The normal fluid energy calculation will be used (IFENER=1), and both the pressure (i.e. mass/momentum) and energy equation systems will be solved at every time step (ISETMO=1 and ISETEN=1, respectively). The specification of IHTWAL(99)=2 indicates that the "full" technique is used to couple the structure and fluid energy equation systems to each other. (With the exception of IHTWAL(99)=2, the values described in this paragraph are the default values and, thus, could have been omitted from the card-image input file.)

The input values associated with time step size selection are described in this paragraph and are all in NAMELIST DATA. The change-based automatic time step size control method is selected by IDTIME=2; default values are accepted for the minimum and maximum time step sizes, the minimum and maximum fractional change per step limits, and the time step size modification factor, since no values are explicitly input for variables DT(3), DT(4), CHGMIN, CHGMAX, and DTFAC, respectively. Until all changes decrease to below CHGMAX to activate the automatic control logic, the time step size will be 0.01 s (=DT(1)) for the first 100 steps (=LASTDT) and then be increased to 1.0 s (=DT(2)).

The iteration control parameters are discussed in this paragraph and are all input in NAMELIST DATA. Rather tight convergence of the inner iteration processes used to solve the pressure and energy equation systems is requested by specifying small values for EPS1, EPS2, and EPS5. The convergence criterion for the pressure equation system will be update every time step since NTHCON(1)=-1. The maximum number of inner interactions (per outer iteration) allowed to obtain convergence of the pressure (ITMAXP) and fluid energy (ITMAXE) equation systems is 249. Since this is a steady-state calculation, the maximum number of outer iterations per time step must be input as 1 in array IT. The program will declare that "steady state has been achieved" when the per step changes in velocity components and fluid enthalpy are less than 10^{-7} (=EPS3) (and the mass residual is below the limits imposed by EPS1 and EPS2).

The input values controlling output are described in this paragraph. The variables to be printed in the first long edit (before taking the first time step) are selected by the input in array

ISTPR; the codes input select volume porosity, surface porosities for all three directions, velocity components for all three directions, temperature, total pressure, density, and volumetric heat source for all fluid cells (displayed by J plane) plus all five thermal structures. Specifying -9999 as the first entry in array NTPRNT indicates that the only other long edit will be produced for the final time step in the run; the variables to be displayed in this other long edit are selected by the input in array NTHPR; these are the same variables selected for the initial long edit except porosities are excluded (since they are constant) and mass residuals are included. Information is written to the plot file every ten time steps, since the first entry in array NTPLOT is -10. Note that the useful information specified in arrays ISTPR, NTHPR, NTPRNT, and NTPLOT is followed by an input of 0 to insure proper processing of the useful input. Other forms of printed output selected are the boundary surface summary (IBSBUG=1 in NAMELIST GEOM), limited amounts of iteration progression information (IMOBUG=1, IENBUG=1, and ITIBUG=2 in NAMELIST DATA), and pump status (IPUBUG=3 in NAMELIST PUMP).

E.3 Printable Output File

The printable output file generated by executing COMMIX-1AR/P on a Cray Research, Incorporated, model X-MP/18 computer using the sample problem card-image input file shown in Table E.1 is reproduced in Table E.3 (which is located after the text of Sec. E.3). The remainder of this section will discuss this output file, working from the beginning to the end. The discussion is divided into three subsections: case summary (Sec. E.3.1), short edit (Sec. E.3.2), and long edit (Sec. E.3.3).

E.3.1 Case Summary

The output begins with a header page; this identifies the code version and the people to contact for user assistance. The next six pages are a listing of the card-image input file. From this point onward, the values of variables shown in the output are either (a) the value actually input or (b) the default value if the variable was not input; the term "input value" will be used to refer to all variable values, whether set by actual input or by default.

The next page tabulates the variables (as input in NAMELIST GEOM) which are used in allocating storage. (Note that the computer program has changed the value of ISTRUC from the input value of 2 to 1.) The important piece of user information here is the minimum size required for array S; in subsequent executions of this same problem, the dimension of array S (as set in subroutine ALTER) may be reduced to this minimum value, which may result in a cost savings on some computer systems. The line after the S array information indicates that the header record has been written on the output restart file. (Additional information would be shown here if ISTBUG=1 in NAMELIST GEOM or ITSBUG=1 in NAMELIST STRUCT.)

The next two and one-half pages provide summaries of the boundary surfaces and the grid system. This begins with the Boundary Surface Summary requested by IBSBUG=1 in NAMELIST GEOM; the two-dimensional I×K display for each J plane shows which faces of each fluid cell are adjacent to boundary surfaces using the character codes displayed in the first part of this block; for example, the character displayed for the fluid cell having location indices ((I,J,K)=) (1,1,1) is "A", which is a code of "101010", which means that the -I, -J, and -K faces of this cell are part of boundary surfaces. The next 4 lines show the actual number of cells and boundary surface elements as counted internally by the computer program. (If the case were to be rerun from ISTATE=0, NL1 and NM1 in NAMELIST GEOM could be reduced to the values shown here; however, the values of NL1 and NM1 must not be input at ISTATE>0.) In the Grid Summary, the DX, DY, and DZ arrays are shown as input; the X, Y, and Z values shown are for the cell-center position of each grid location. The Surface-Surface Element Summary shows information for each surface as defined in the Boundary Surface Identification input block and the components of the unit surface-normal vectors input using arrays XNORML, YNORML, and ZNORML in NAMELIST GEOM.

The Flag Summary is a tabular listing of most integer constants input in NAMELISTs GEOM and DATA. This is followed by a summary of parameters associated with the change-based automatic time-step-size controller since IDTIME=2.

The next one and one-quarter pages provide summaries of the fluid properties. There is one block of output for each of the three fluids. The property values are displayed for the

temperature range FCTLO to FCTHI (as input in NAMELIST DATA). Since IFPROP=0, "detailed" properties are used for fluid number 1; the string "SODIUM" in the header confirms that the routines which describe sodium have been properly linked with the rest of the computer program. For the other two fluids, the correlations being used to calculate "simplified" properties are shown. The fluid properties are followed by a summary of the correlations being used for the properties of the two structure materials and for the seven fluid/structure heat transfer correlations; the column headings show the array names by which the constants were input in NAMELIST DATA..

The Boundary Conditions Summary shows the values for all of the arrays in NAMELIST DATA associated with boundary conditions. The Transient Functions Summary has a block for each of the two transient functions input; shown for each function are the results of the cubic-spline fitting process for each range of the function followed by tabulations of the function value at fifty-one equally-spaced time values spanning the range input for TVAL. The next output block tabulates the number of cells of each fluid type.

The Pump Model Summary begins with a line showing the parameters for each pump. This is followed by a summary of the head and torque correlations being used; the values shown for arrays CORPH8 and CORPT8 are the defaults since these arrays were not input; the values in the rightmost column will be used for array CORPT8 since IFPUMP=4. The final line of this block shows the pump location indices (" 20207" is location (2,2,7)) and orientation (where IDIR values of 1, 2, and 3 correspond to directions I, J, and K, respectively).

The Force Structure Summary begins by showing the information input for each of the seven prototypes defined in NAMELIST DATA. This is followed by a tabulation of the six friction factor correlations which were input in NAMELIST DATA. This block concludes with a statement of how many momentum cells were assigned force structures in the Force Structure Location input block.

The Thermal Structure Summaries section has several parts. The first part is one line per thermal structure prototype showing key values; the first number on each line is the order

in which the prototypes were defined in the Thermal Structure Prototype input block and the second number is the value specified for variable N in NAMELIST T for this prototype; the third and fourth columns show the number of fluid cells adjacent to each side of the prototype as input in the Thermal Structure Location input block; the fifth through eighth columns show the heat transfer correlation indices (IHT) and characteristic lengths (HYD) as input in NAMELIST F; the last three columns show the multiplicity factor (RODFR), the geometry type (IXYZ), and the heat-source transient function number (NT) as input in NAMELIST T. The next part is an edit consisting of two subblocks for each thermal structure. The first subblock shows the layer type construction of the prototype, working from the outside to the inside; this edit will be explained using prototype number 1 as an example; the line beginning with "TYPE" indicates the beginning of a prototype and the number on this line is the prototype number ("1"); the appearance of "FLUID" on the next line indicates that fluid cells interact with the outside surface of this thermal structure; the line beginning with "MATERIAL" indicates that the outermost material region has one partition, is composed of material number 2, has no heat source, and is 0.457 mm thick; the appearance of "GAP" on the next line indicates a change in material regions, showing the gap heat transfer coefficient ($2 \times 10^5 \text{ W/m}^2 \cdot ^\circ\text{C}$) and thickness (0.083 mm); the next five appearances of "MATERIAL" indicate that the nextmost inner material region has five partitions, each of which is composed of material number 1, has a heat source (539.04 MW/m^3), and is 0.555 mm thick; the final line shows the total outside and inside surface area [m^2] for the full multiplicity of the thermal structure summed over all fluid cells to which it is attached; the absence of a "FLUID" line after the final "MATERIAL" line indicates that fluid cells do not interact with the inside surface of this thermal structure. (The surface area and volume values shown on the "MATERIAL" lines are not very useful; they are a "per element" type number and must be multiplied by a geometry factor, which depends on IXYZ and RODFR, to get a true area and volume.) The second subblock shows the volume, mass, and heat capacity (mass times specific heat capacity) by material type for this thermal structure; the mass and heat capacity values shown here are computed at temperature TEMPO(1), as input in NAMELIST DATA; note that the gaps separating material regions have volume but no mass. After these subblocks are printed for all thermal structures, a final "Inventory" block is printed, showing the mass, volume, and heat capacity for each material summed over all thermal structures.

The Summary of Radiation Heat Transfer section begins with a tabulation of the values for the nonarray variables which were input in NAMELIST DATA. The next table has one line per thermal structure, showing which surfaces of which thermal structures were defined to be radiation surfaces based on the values specified for KRAD in NAMELIST M. The next table presents this same information but is ordered by radiation surface number. The final table in this section is the view factor array (FVIEW(n,m), as input in NAMELIST DATA); only the useful portion of this array is printed (i.e. for m and n values ranging from 1 up through 4 (NRAD)); the user will want to examine this table carefully to insure that this two-dimensional array was input properly.

The next block of output is the initial density for each fluid; this is computed from the fluid properties functions at temperature TEMPO(n) and pressure PRES0(n) for each fluid number n; this is the density value used by the computer program to establish the gravity-head-based reference-state pressure distribution.

The Surface Area Summary shows the area of each boundary surface. This table is followed by a line showing the total surface area and the total fluid volume.

The Turbulence Model Summary shows constants associated with the fluid turbulence model. For ITURKE=0, the only significant items in this table are the turbulent viscosity (TURBV) and the turbulent thermal conductivity (TURBC), both of which were input in NAMELIST DATA.

This next section of output is a "long edit" showing values of many variables before the first time step is taken. In the interest of compactness, only the Global Balances section of this initial long edit has been reproduced in Table E.3; the rest of the long edit (variables selected by array ISTPR in NAMELIST DATA) has been deleted. The format of a long edit is described in Sec. E.3.3.

The Control Summary follows the initial long edit. It provides a compact summary of most nonarray floating-point variables input in NAMELISTs GEOM and DATA. Most of these variables are related to time-step-size selection and iteration control.

E.3.2 Short Edit

There is at least one line of printed output produced for each time step; this is the line which begins with the character "#" and is denoted a "short edit". The items on this line are identified by a header line ("NT TIME ..."), which is printed every fifty steps. The first ten numbers on the short edit line are always printed and are as follows: (1) NT: time step number, (2) TIME: case time [s], (3) DTIME: time step size [s], (4) IT: the number of outer iterations used for this step, (5) MAX(DL)/DCONV: the maximum mass residual divided by DCONV [—], (6) MAX(DU)/VELMAX: the maximum change in the I-direction velocity components during this step divided by VELMAX [—], (7) MAX(DV)/VELMAX: the maximum change in the J-direction velocity components during this step divided by VELMAX [—], (8) MAX(DW)/VELMAX: the maximum change in the K-direction velocity components during this step divided by VELMAX [—], (9) MAX(DH/H): the maximum relative change in the fluid enthalpy during this step [—], and (10) TLEFT: the CPU time remaining to complete the run [s]. The other three items are only printed for the steps selected by array NTHCON in NAMELIST DATA; they are as follows: (11) COURDT: Courant-based time-step-size limit [s], (12) DCONV: value (cf. Eq. 12.10) below which the mass residuals must be for the pressure equation system to be considered converged [kg/s · m³], and (13) VELMAX: the maximum fluid speed [m/s] (cf. denominator of Eqs. 12.17 through 12.19).

The additional information shown for each time step is optional, being selected by the various "xxxBUG" input variables nonzero. There are five such optional groupings shown in the sample problem output.

The line beginning with "NP=" and the next two lines are associated with the pump modeling and are printed (once per outer iteration) since IPUBUG ≥ 2 in NAMELIST PUMP. The meanings of these variables are as follows: NP: the pump number for which information is displayed in these three lines; PUMPH: the current head being supplied [m]; PUMPHR: the reference pump head [m]; TORQ: hydraulic torque [N · m]; TORQR: reference hydraulic torque [N · m]; TORQF: frictional torque [N · m]; TORQFR: reference frictional torque [N · m], which is not used for IFPUMP=4; TORQMT: motor torque [N · m], which is only defined for

$ISTATE \geq 2$; TORQM: reference motor torque [$N \cdot m$]; DENRO: product of density and gravitational acceleration [$kg/m^2 \cdot s^2$]; QCUT: relative flow rate below which linear head versus flow rate relationship will be used [-]; DPPUMP: pressure increase ($=PUMPH \cdot DENRO$) being supplied [Pa]; PSSR: relative speed [-] ($=PUMPS/PUMPSR$); QFSR: relative volumetric flow rate [-] ($=QFLOW/QFLOWR$); XPUMPH: operating point [radian] in $\tilde{Q}-\tilde{N}$ plane (X as defined by Eq. 7.4); HTOT: head function [-] ($WH(X)$ as defined by Eq. 7.9 or 7.16); and XTORQT: hydraulic torque function [-] ($WT(X)$ as defined by Eq. 7.5 or 7.12). (Of the preceding values, the following were input in NAMELIST PUMP: PUMPHR, TORQR, TORQM, DENRO, QCUT, PUMPS, PUMPSR, and QFLOWR.)

The line beginning with "C.G.:" is information associated with the conjugate-gradient-based solution of the pressure equation system and is printed (once per outer iteration) since $IMOBUG \geq 1$ in NAMELIST DATA. The values printed are all for the current outer iteration and are as follows: ITR: number of inner iterations; DLMAX: maximum mass residual [$kg/s \cdot m^3$]; DCONV: value (cf. Eq. 12.10) below which the mass residuals must be for the pressure equation system to be considered converged [$kg/s \cdot m^3$]; FACTORIZED: "YES" (or "NO") indicates that a new factorization (incomplete Cholesky) of coefficient matrix was (or was not) computed; and CPU: central processor time [s] spent solving the pressure equation system.

The line beginning with "... MOLOOP" is information associated with the solution of the momentum/mass (i.e. pressure) equation system and is printed (once per outer iteration) since $ITIBUG \geq 2$ in NAMELIST DATA. The values printed are all for the current outer iteration and are as follows: ITMO: not currently used (will always be "1"); DU...AT: maximum change in I-direction velocity components during this outer iteration divided by $VELMAX \cdot OMEGAV$ [-] and the cell location at which the maximum occurs ("10102" is cell location (1,1,2)); DV...AT: same as "DU...AT" but for J-direction velocity components; DW...AT: same as "DU...AT" but for K-direction velocity components; ITP: number of inner iterations; and DLMAX...AT: maximum mass residual [$kg/s \cdot m^3$] and the cell location at which the maximum occurs. Note that the changes in velocity components printed in this line are the changes between this outer iteration and the previous outer iteration of this time step, whereas those printed on the line beginning with "#" are the changes between this time step and the

previous time step; in this steady-state calculation with one outer iteration per time step ($IT = 1$) and $OMEGAV = 1$, the changes shown in these two lines are the same; they will be different during a transient calculation with more than one outer iteration per time step.

The line beginning with `..ENLOOP` is information associated with the solution of the fluid energy equation system and is printed (once per outer iteration) since $ITIBUG \geq 2$ in `NAMELIST DATA`. The values printed are as follows: `DHOHT`: maximum relative change in fluid enthalpy during the current outer iteration [–]; `IJKT`: cell location for which `DHOHT` is printed ("50301" is cell location (5,3,1)); `HL`: enthalpy [J/kg] for fluid cell location `IJKT` at the end of the current outer iteration; `HLT`: enthalpy [J/kg] for fluid cell location `IJKT` at the end of the prior outer iteration; `DHOH`: maximum relative change in fluid enthalpy [–] during the time step; `IJK`: cell location for which `DHOH` is printed ("50301" is cell location (5,3,1)); and `HL`: enthalpy [J/kg] for fluid cell location `IJK`. (Again, the change between successive outer iterations is the same as the change between successive time steps in this steady-state calculation.)

The lines beginning with `"N="` are information associated with the radiation heat transfer surfaces and are printed (once per time step) since $IENBUG \geq 1$ in `NAMELIST DATA`. There is one line printed for each of the `NRAD` ($= 4$) radiation surfaces and the values printed are as follows: `N`: radiation heat transfer surface number; `TROD`: temperature [$^{\circ}\text{C}$]; `QRN`: net radiation heat transfer rate [W] to this surface (a "minus sign" indicating net heat transfer from this surface) at the end of this time step; `QROLD`: same as `QRN` but at the beginning of this time step; `AREATR`: area [m^2]; and `OMR`: underrelaxation factor [–] (as input in `NAMELIST DATA`). The present time step was performed using the heat transfer rates shown as `"QROLD"`; the `"QRN"` values will be used during the next time step.

The short edits and intervening lines are shown in Table E.3 for the first few and the last few time steps; the output from the intervening time steps have been deleted (denoted by `"== => Deleted short edits (68 pages) for time step numbers 22 through 312. <== ="`.) Several other useful pieces of information are discussed before discussing the "long edit".

Between time step numbers 7 and 8, a line is printed which reads *****TIMSTP: AUTO TIME STEPS INITIATED**". This line is printed since **IDTIME=2** and the maximum per step changes in kinetic energy, fluid temperature, and turbulent viscosity are less than **CHGMAX** for the first time in the calculation. The time step size will be adjusted automatically during the remainder of the steady-state calculation to keep the maximum changes in these three quantities between **CHGMIN** and **CHGMAX** (both of which are input in **NAMELIST DATA**).

Between time step numbers 10 and 11, two lines are printed which read **"UNFORMATTED FILE OPENED ON UNIT 76"** and **"PLOTING INFORMATION WRITTEN TO UNIT 76"**. These two lines are printed since **NTPLOT=-10** was input and the tenth time step has been reached. They indicate that the binary-format plot file has been opened, the header records have been written, and the data records have been written for time step number 10. Data records will continue to be written to the plot file every ten time steps; however, the writing of the additional records will not be noted in the printable output file.

Between time step numbers 18 and 19, a line is printed which reads *****AMAIN: DTL AND DEK HAVE BEEN LESS THAN CHGMIN FOR 5 STEPS; INCREASE THE TIME STEP SIZE TO 2.00000E-02**". This line is printed since **IDTIME=2**, **IMOBUG $\geq$ 1** or **ITIBUG $\geq$ 1**, and the maximum per step changes in kinetic energy, fluid temperature, and turbulent viscosity are less than **CHGMIN** for five consecutive time steps (for the first time in the calculation). The time step size will be increased (by a factor of **DTFACT**) to 0.02 s for the next time step. Note that the short edit for time step number 19 shows the increased time step size. The user has no control over the number of consecutive steps which must transpire with changes less than **CHGMIN** in order to increase the time step size; the initial value is 5; the number will be doubled if the computer program must reduce the time step size to keep changes below **CHGMAX** immediately following an increase in time step size.

The "time-step-increase" line is immediately followed by a line which begins **"CONTROL CHANGE"**. This line is printed every time (i.e. not a function of **IDTIME**, **IMOBUG**, or **ITIBUG**) that the time step size or the allowed maximum number of outer iterations per time step changes. The value of **NTIME** is the time step number at which the

changes occur; the rest of the line contains the new values of the following quantities: DTIME: time step size [s] for fluid calculations; DTFUEL: time step size [s] for thermal structure energy calculations; DTWALL: time step size [s] for duct wall energy boundary conditions; DTENER: not currently used; and ITERMX: maximum number of outer iterations per time step. During a steady-state calculation, the values of DTFUEL and DTWALL are set by the computer program to an extremely large value (i.e. 10^{40}) to force the thermal structures and duct walls to remain in thermal equilibrium with the adjacent fluid cells; during a transient calculation, both of these two quantities will be set by the computer program to the value of DTIME.

Following the short edit for time step number 319 is a line reading "STEADY STATE HAS BEEN ACHIEVED TO WITHIN EPS3 = 1.000E-07". This means that the step-to-step changes in the three velocity components and the enthalpy (all on a relative basis) are less than EPS3 and the mass residual is less than DCONV for all fluid cells and, therefore, the computer program is terminating the calculation. Although this banner is a good indication that steady state has been reached, the user is encouraged to check the upcoming long edit to insure that, indeed, steady state has been achieved to the user's satisfaction, since not all variables are included in the program's check for steady state.

Following the "STEADY STATE" banner, there is printed a table entitled "TIMING SUMMARY". The values in this table are CPU times [s] for this computer run. The first line of this table is the total amount of time consumed performing calculations but excluding the time spent processing input and output. The other items in the table are breakdowns of the major contributors to this total. The timings are subdivided into (a) construction time and (b) solution time for the four major equation systems: momentum (lines 2 and 3), fluid energy (lines 4 and 5), turbulent kinetic energy (lines 8 and 9), and turbulent kinetic energy dissipation rate (lines 10 and 11). The other two lines are thermal structures (line 6, although there is additional thermal structure time included in the time shown for "ENERGY EQ CONSTRUCTION") and radiation heat transfer (line 7). The sum of lines 2 through 11 will always be less than the value shown on line 1, which, in turn, will always be less than the total time consumed by the computer program.

Following the "timing summary", there are four lines of output concerning the restart file. The first line indicates that an output restart file has been written (as requested by `IFRES=1` in `NAMelist GEOM`), showing the case time, outer iteration ("0" indicates that the time step was complete), and time step number. The next two lines are the lengths of the various blocks on the restart file. These three lines will be printed by the computer program when this restart file is input to a subsequent `COMMIX-1AR/P` computer program run and may be checked by the user to insure that the correct file was used. The fourth line indicates that plot-file-type records have been appended to the output restart file; if an output restart file is requested, these records are always appended; they are not read by the `COMMIMX-1AR/P` computer program but are, instead, read by the graphics postprocessor programs `VECT` and `TEMP` (cf. Appendix D).

Following the printed lines associated with the output restart file is a final long edit. This long edit is produced because steady state was achieved in this computer run. (If steady state had not been achieved, a final long edit would have still been printed since "`NTPRNT=-9999`" was input in `NAMelist DATA`.) The format of a long edit is described in the next subsection.

E.3.3 Long Edit

There are five "times" at which a "long edit" is produced: (1) before the first time step in a steady-state calculation, (2) when steady state is achieved, (3) before the first time step in a transient calculation, (4) at the time step numbers specified in array `NTPRNT` in `NAMelist DATA`, and (5) at the times specified in array `TPRNT` in `NAMelist DATA`. The first three of these are selected automatically by the computer program; the other two are selected by the user. Each long edit consists of two major sections: nonoptional and optional; the items shown in the nonoptional section are always printed; the items in the optional section are selected by the user, using array `ISTPR` for edits produced at "time" items 1 and 3 above and using array `NTHPR` for "time" items 2, 4, and 5. The nonoptional and optional sections are described below.

The nonoptional section of a long edit begins with the heading "GLOBAL BALANCES AT TIME= ...". The first subblock is a summary of boundary surfaces, having one line for each boundary surface. The quantities shown for each boundary surface are as follows: (1) the boundary surface number; (2) the mass convection flow rate [kg/s]; (3) the energy convection rate [W] as a result of mass convection; (4) the energy conduction rate [W]; (5) the average velocity [m/s] in the direction of the unit surface-normal vector; (6) the average fluid enthalpy [J/kg]; (7) the average (area-weighted) temperature [$^{\circ}\text{C}$]; and (8) the bulk (mass-flow-rate-weighted) temperature [$^{\circ}\text{C}$] for surfaces having nonzero mass convection. A positive value of convection and conduction quantities indicates addition to the fluid system. At the end of a steady-state calculation, the user should check for balances between inlet and outlet mass flow rates for each region (e.g. boundary surface number 20 versus 21 and boundary surface number 26 versus 27) and that mass flow rates shown for expansion cells are small (e.g. boundary surface numbers 7 and 15).

The boundary surface summary is followed by a section entitled "MASS BALANCE". The values [kg/s] shown in the three lines of this table are as follows: MASS ACCUMULATION RATE (DMASDT): the derivative with respect to time of cell fluid mass, summed over all fluid cells; NET MASS CONVECTED INTO CONFIGURATION (FLOSUM): sum of "MASS CONVECTED" column in above boundary surface summary; GLOBAL CONSERVATION OF MASS: line one minus line two; DLSUM: product of mass residual and cell fluid volume, summed over all fluid cells. During a steady-state calculation, the derivative of cell fluid mass with respect to time is defined as zero; therefore, the value of DMASDT is zero; when steady state has been achieved, the values of FLOSUM and DLSUM should both be small (at least on a relative basis when compared with the gross inlet or outlet mass flow rate). (The results in Table E.3 show a mass flow rate imbalance of only 4 mg/s relative to a gross flow rate of 572 kg/s.) During a transient calculation, the values of DMASDT and FLOSUM will not necessarily be small, but the GLOBAL CONSERVATION OF MASS (along with DLSUM) should be small for a well-convergence solution.

The mass balance is followed by a section entitled "FLUID ENERGY BALANCE". The values [W] shown in the six lines of this table are as follows: ENERGY ACCUMULATION

RATE (DEDT): the derivative with respect to time of the cell fluid energy content, summed over all fluid cells; **NET ENERGY CONVECTED INTO CONFIGURATION (EINSUM):** sum of "ENERGY CONVECTED" column in above boundary surface summary; **NET ENERGY CONDUCTED INTO CONFIGURATION (QINSUM):** sum of "ENERGY CONDUCTED" column in above boundary surface summary; **ENERGY TRANSFERED TO/GENERATED WITHIN CONFIGURATION (QSOURC):** sum of net heat transfer to (and generation in) fluid; **ENERGY EXCESS DUE TO MASS RESIDUAL (EDLSUM):** product of mass residual, enthalpy, and cell fluid volume, summed over all fluid cells; **GLOBAL CONSERVATION OF ENERGY:** line one minus the sum of lines two through five. During a steady-state calculation, the derivative of fluid cell energy content with respect to time is defined as zero; therefore, the value of DEDT is zero; when steady state has been achieved, the values of EDLSUM and the global conservation of energy should both be small (at least on a relative basis when compared with the gross inlet or outlet energy flow rate and the source); under these conditions, the energy generated internally is being removed by convection and conduction at the boundaries. (The results in Table E.3 show an energy rate imbalance of only 357 W relative to a gross generation of 106 MW.) During a transient calculation, the value of DEDT will not necessarily be small, but the GLOBAL CONSERVATION OF ENERGY (and the value of EDLSUM) should be small for a well-convergence solution.

Following the fluid energy balance is a table with the heading "FLUID INVENTORY". There is one line for each fluid. Shown for each fluid are the following: (1) fluid number (n); (2) total mass [kg] (product of density and fluid volume, summed over all fluid cells); (3) total energy content [J] (product of enthalpy, density, and fluid volume, summed over all fluid cells); (4) average enthalpy [J/kg] (column 3 divided by column 2); and (5) average temperature [°C] (from the equation of state routines for fluid number n using the enthalpy in column 4 and pressure PRES0(n) as input in NAMELIST DATA). The final line is the total energy of all fluid types.

The fluid inventory is followed by a table entitled "THERMAL STRUCTURE HEAT BALANCE". There is one line for each thermal structure prototype. The following quantities, summed over all multiplicities and replications, are shown for each thermal structure:

(1) thermal structure index number; (2) gross heat generation rate [W] within the structure (i.e. the Q input in NAMELIST M); (3) the heat transfer rate [W] from the outer surface of the structure to the fluid (a minus sign indicates heat transfer from the fluid to the structure; radiation heat exchange with other thermal structures is not included); (4) same as column 3 but for the inside surface; (5) sum of columns 3 and 4; (6) net heat source rate [W] for the structure (column 2 minus column 5; since columns 3 and 4 exclude radiation heat transfer, the number in column 6 is a true net heat source only if this thermal structure does not exchange energy via radiation); (7) the amount of stored energy [J] in the structure above a temperature reference level of TEMPO(1); (8) the average temperature [$^{\circ}\text{C}$]. The final line shows a total or average value for the collection of all thermal structures. After a steady-state calculation has converged, the user will want to check these entries to insure that all heat being generated in thermal structures is being transferred to fluid.

The thermal structure heat balance is followed by a table entitled "RADIATIVE SURFACES". There is one line for each radiation surface showing (1) the radiation surface number, (2) the temperature [$^{\circ}\text{C}$], and (3) the net radiation heat transfer rate [W] to the surface (a minus sign indicates net heat transfer from the surface). The sum of the heat transfer rates in column 3 should be zero; if not, check the consistency of the view factors (array FVIEW in NAMELIST DATA).

Several examples are provided at this point to show the interrelation among various quantities in the long edit described above. (1) Thermal structure number 1 represents the fuel pins; the internal heat generation rate is 106.2361 MW, and heat is given to the fluid (primary sodium) adjacent to the outside surface at this same rate, as shown in the thermal structure heat balance. (2) Thermal structure number 2 represents the IHX tubes; they receive heat from the fluid (primary sodium) adjacent to the inside surface at a rate of 106.0476 MW and give heat to the fluid (intermediate sodium) adjacent to the outside surface at the same rate, as shown in the thermal structure heat balance; this rate is the same as the difference between the IHX intermediate sodium outlet (523.8843 MW for boundary surface number 27) and inlet (417.8367 MW for boundary surface number 26) energy convection rates. (3) Thermal structure number 3 represents the RV wall; it receives heat from the fluid (primary sodium) adjacent to

the inside surface at a rate of 0.1885 MW and gives heat to the fluid (argon) adjacent to the outside surface at a rate of 0.0048 MW, as shown in the thermal structure heat balance; since the outside surface of thermal structure number 3 is radiation surface number 1, the remainder of the heat transfer at the outer surface is by radiation at a rate of 0.1837 MW, as shown under radiative surfaces. (4) Thermal structure number 4 represents the GV wall; its inside surface receives heat from the adjacent fluid (argon) at a rate of 0.0048 MW and, since the inside surface of thermal structure number 4 is radiation surface number 2, receives additional heat by radiation (from radiation surface number 1) at a rate of 0.1837 MW; its outside surface gives heat to the adjacent fluid (air) at a rate of 0.1130 MW and, since the outside surface of thermal structure number 4 is radiation surface number 3, gives up additional heat by radiation at a rate of 0.0755 MW; the net rate of heat transfer on each side of the GV wall is 0.1885 MW (which is the same rate as for the RV wall). (5) Thermal structure number 5 represents the CV wall; its inside surface receives heat by radiation (from radiation surface number 3) at a rate of 0.0755 MW, since the inside surface of thermal structure number 5 is radiation surface number 4; its inside surface gives heat to the adjacent fluid (air) at this same rate; there is no heat transfer at the outside surface of the CV. (6) The air flowing through the GV/CV gap receives heat from the outside surface of thermal structure number 4 (0.1130 MW) and the inside surface number 5 (0.0755 MW); this total rate (0.1885 MW) is the same as the difference between the air outlet (1.5504 MW for boundary surface number 21) and inlet (1.3623 MW for boundary surface number 20) energy convection rates (at least within the tolerance of there being a 0.0004 MW global energy rate imbalance for the fluid system). (7) The rate of energy generation in the fuel pins (106.2361 MW) is removed from the system by the combined convection rates of the intermediate sodium (106.0476 MW) and the air (0.1885 MW).

This concludes the nonoptional section of a long edit. The remainder of the long edit displays various planes of variables and thermal structure details as selected by the user via the information input using arrays ISTPR and NTHPR in NAMELIST DATA; the variables are displayed in the order in which they are requested in arrays ISTPR and NTHPR. Since this is not the first long edit in a steady-state or transient calculation, the information shown is governed by NTHPR. Several portions of this output will be discussed.

The first entry in array NTHPR is "-01201"; since this value is negative, it is used in conjunction with the second entry, whose value is "01203". The combination of these two entries requests the I-direction component of velocity (UL) to be printed for J planes 1 through 3. This produces the first three groupings of optional printout, entitled "UL (M/S) : VELOCITY IN X-DIRECTION AT TIME= 4.067E+06 SECONDS". There is one block of output for each of the three J planes. For each J plane, the display is two dimensional, with the I spatial index running across the page and the K spatial index running down the page; these columns and rows are labeled. The velocity shown for each position is that at the +I face of the cell at position (I,J,K). If this cell face lies on a boundary surface, the value will be shown as "0.0" in this display; velocities on boundary surfaces must be requested using a code of "901nn", where "nn" is a two-digit boundary surface number, in array NTHPR (or ISTPR). The value will also be shown as "0.0" for grid positions where there is no fluid cell (e.g. location (5,1,12)).

The next pair of codes in array NTHPR ("-02201,02202") requests similar displays for the J-direction velocity components. These input values produce the next 2 blocks of output, entitled "VL (M/S) ...". The value displayed for a J-direction velocity component is the value at the +J face of the cell. Note that the input in array NTHPR does not request values to be displayed for J-plane number 3, since all of the +J cell faces in this plane lie on boundary surfaces.

The next pair of codes in array NTHPR ("-03201,03203") requests similar displays for the K-direction velocity components. These input values produce the next 3 blocks of output, entitled "WL (M/S) ...". The value displayed for a K-direction velocity component is the value at the +K face of the cell.

The next five pairs of codes in array NTHPR ("-05201,05203", "-08201,08203", "-07201,07203", "-09201,09203", and "-15201,15203") request similar displays for the fluid temperature, pressure, density, mass residual, and volumetric heat source. These input values produce the next 5*3 blocks of output, entitled "TL ...", "PRESSURE ...", "RL ...", "DL ...",

and "QSOUR ...", respectively. The value displayed for all of these variables is the cell-center value for the cell at location (I,J,K). The value of "0.0" will be shown for grid locations where there is no fluid cell (e.g. I=7).

The last pair of codes in array NTHPR ("−80001,80005") requests displays of all five thermal structures. The format of the display depends on whether one or both surfaces of the thermal structure interact with fluid cells. Example of these two cases are discussed in the next two paragraphs.

Thermal structure number 1, which represents the fuel pins, is an example of a thermal structure for which only one surface interacts with fluid cells. There is one line of output for each thermal structure element. Column one of the output shows the thermal structure element number (as assigned by the computer program based on the order of records in the Thermal Structure Prototype and Location input blocks). The next three columns are for the surface which interacts with fluid cells, showing the fluid cell location indices (a six-digit integer of the form "iijjkk", where ii, jj, and kk are two-digit integers indicating, respectively, the I, J, and K spatial location indices), the temperature [°C] in that fluid cell, and the heat transfer coefficient [$\text{W}/\text{m}^2 \cdot ^\circ\text{C}$] between the surface of the thermal structure and the adjacent fluid cell. The remainder of the display shows the temperature [°C] at the center of each partition, working from the outside to the inside; in this example, the first partition is the cladding and the other five partitions are in the fuel; this printing of temperature will be continued to additional lines, if needed, to show temperatures for all partitions.

Thermal structure number 2, which represents the IHX tubes, is an example of a thermal structure for which both surfaces interact with fluid cells. The output format is similar to the preceding example. There is one line of output for each thermal structure element, and column one of the output shows the thermal structure element number. The next three columns are for the outside surface, showing the fluid cell location indices, the temperature [°C] in that fluid cell, and the surface heat transfer coefficient [$\text{W}/\text{m}^2 \cdot ^\circ\text{C}$]. The next three columns show the same information for the inside surface. The remainder of the display shows the

temperature [$^{\circ}\text{C}$] at the center of each partition, working from the outside to the inside; in this example, all three partitions are in the same material region; this printing of temperature will be continued to additional lines, if needed, to show temperatures for all partitions.

This concludes the printing of all of the optional items which were requested using codes in array NTHPR. Optional displays of boundary surface variables were not requested in this sample problem; the two-dimensional displays of these quantities are similar to those described above for internal cells.

A few comments are given at this point on the meaning of these results. (1) The collection of numbers on the global balances part of the final long edit provides a degree of confidence that steady state has actually been achieved in this calculation; however, some of the conditions differ from those expected or desired. (2) The volumetric flow rate through the pump is about 8% higher than desired (as shown by $\text{QFSR}=1.083$ in the short edit for the pump for the final time step); the user can alter this flow rate by adjusting the pump operating point and/or the frictional resistances within the RV (and rerunning the calculation). (3) The temperature rise across the fuel assemblies is about 8% lower than expected; this is a direct result of the flow rate being higher than desired. (4) The temperature of the cold pool in the RV is about 40°C higher than desired; this is primarily due to the inaccuracies associated with the first-order finite-difference formulation when applied to the IHX which only has three axial nodes; this error can be reduced by using more axial nodes along the length of the IHX (and this is recommended). Further discussion of these results is provided in Sec. E.4.

Table E.3 Listing of Printable Output File

The next 40 pages are a reproduction of the printable output file for the sample problem.

A Three-Dimensional Transient Single-Phase Computer Program for Thermal Hydraulic Analysis of Single and Multicomponent Systems

developed in the

Reactor Analysis Division
Argonne National Laboratory

under sponsorship of the
United States Department of Energy

contact
phone

R. N. Blomquist	708-252-8423
P. L. Garner	708-252-4872
E. M. Gelbard	708-252-4831

at

Building 208
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439

Updated to Version 910515 (09/18/91)

```

Mini-mod (8,3,12): 90-degree sector of modular pool-type LMR:
  Includes code+IHX(pri)+pump+IHX(int)+RVACS but not to scale.
  Steady-State: full-power(106MW), full-flow.
  Flow path for sodium within reactor vessel(RV) is (I,K,J):
    up thru fuel assemblies (1,1-3,4-6) with heat source in K=5
    to hot pool (1,1-3,7-9)+(1-2,1-3,10-11)
    down thru IHX (2,3,3-9) with heat transfer in K=7-9
    up thru pump (2,2,3-9) with DELTA-P supplied at K=7
    down thru pipe (2,1,3-9)
    to cold pool (1-2,1-3,1-2)+(1,1-3,3)
    to fuel assemblies (closed loop).
  Stagnant argon between RV and guard vessel(GV) (3-4,1-3,1-11).
  Flowing air between GV and confinement vessel(CV) (5-6,1-3,1-11).
  Radiation heat transfer between RV, GV, and CV.
  (No cells in radial node 1=7.)
  Intermediate side of IHX in (8,3,6-10) with heat transfer in K=7-9.
  Expansion Cells: primary-sodium=(2,1,12), argon=(4,1,12).
  Boundary Surfs.: 1=8-pri.Na; 9-15=argon; 16-21=air; 22-27=int.Na;
    28-33=zero.porosity.internals.
  Thermal Structs.: 1=fuel; 2=IHX; 3=RV; 4=GV; 5=CV.
  Fluids: #1=sodium(pri+int); #2=argon; #3=air.
  Solids: #1=fuel(metal); #2=steel.
  Force Structs.: 1=3-core; 4=IHX(pri); 5=IHX(int); 6=pump; 7=pump.
  Heat Transf. Correls.: 1=core; 2=unused; 3=IHX(pri); 4=IHX(int);
    5=RV/GV(gap); 6=GV/CV(gap); 7=RV(int);
  Trans. Funct.: 1=core power; 2=pump torque.
  Turbulence: user-input constant.
  S array: 71749 on short-word, 64405 on long-word machine.
  Revised 02.18/92, P.Garner, PIGarner@anl.gov, 708-252-4872
&GEOM
  IGEOM=-1, NL1=482, NM1=205,
  ISYMCH=5, IFITEN=3, IFITX=3, IFREB=0, IFRES=1,
  ITURKE=0, NFORCE=7, ISTRUC=2, ISTBUG=0, IBSBUG=1,
  IMAX=8, JMAX=3, KMAX=12, NSURF=33,
  DX=1.55,1.30,2.076,2.171,.161,.750,
  DY=2*.3927,.7854,
  DZ=2*.15,.51,1.26,1.73,1.42,3*1.78,2*1.06,1.0,
  XNORML=1.,-1.,4*0., 0.,-1., 1.,-1.,5*0., 1.,-1.,4*0.,
  1.,-1.,4*0., 1.,-1.,4*0.,
  YNORML=2*0.,1.,-1.,2*0., 2*0., 2*0.,1.,-1.,3*0., 2*0.,1.,-1.,2*0.,
  2*0.,1.,-1.,2*0., 2*0.,1.,-1.,2*0.,
  ZNORML=4*0.,1.,-1., -1.,0., 4*0.,1.,2*-1., 4*0.,1.,-1.,
  4*0.,1.,-1., -1.,0.,1.,-1.,
&END
  REG -1. 1 1 1 3 1 11 1 RV:R+(CTR)
  REG -1. 2 2 1 1 12 12 1 RV(EXP):R+
  REG -1. 2 2 1 2 3 9 2 RV:R-(NO.Q)
  REG -1. 2 2 1 1 12 12 2 RV(EXP):R-(NO.Q)
  REG -1. 1 1 1 1 1 3 3 RV:O+
  REG -1. 1 1 1 1 7 11 3 RV:O+
  REG -1. 2 2 1 1 1 1 3 RV:O+
  REG -1. 2 2 1 1 10 11 3 RV:O+
  REG -1. 2 2 1 1 12 12 3 RV(EXP):O+
  REG -1. 1 1 3 3 1 3 4 RV:O-
  REG -1. 1 1 3 3 7 11 4 RV:O-
  REG -1. 2 2 3 3 1 6 4 RV:O-
  REG -1. 2 2 3 3 10 11 4 RV:O-

```

```

REG -1. 2 1 1 12 12 4 RV(EXP;Z-
REG -1. 1 1 3 1 5 RV:Z+
REG -1. 1 1 3 11 11 5 RV:Z-
REG -1. 2 2 3 11 11 5 RV:Z-
REG -1. 2 2 1 1 12 12 7 RV(EXP;Z-
REG -1. 2 2 1 3 1 2 8 RV:R-(Q)
REG -1. 2 2 3 3 9 8 RV:R-(Q)
REG -1. 2 2 1 3 10 11 8 RV:R-(Q)
REG -1. 3 3 1 11 9 RV/GV:R+
REG -1. 4 4 1 1 12 12 9 RV/GV:R+(EXP)
REG -1. 4 4 1 3 1 11 10 RV/GV:R-
REG -1. 4 4 1 1 12 12 10 RV/GV:R-(EXP)
REG -1. 3 4 1 1 11 11 RV/GV:O+
REG -1. 4 4 1 1 12 12 11 RV/GV:O+(EXP)
REG -1. 3 4 3 1 11 12 RV/GV:O-
REG -1. 4 4 1 1 12 12 12 RV/GV:O-(EXP)
REG -1. 3 4 1 3 1 13 RV/GV:Z+(NOFLOW)
REG -1. 3 3 1 11 14 RV/GV:Z-(NOFLOW)
REG -1. 4 4 2 3 11 14 RV/GV:Z-(NOFLOW)
REG .059375 4 4 1 1 12 12 15 RV/GV:Z-(EXP)
REG -1. 5 5 1 3 1 11 16 GV/CV:R+
REG -1. 6 6 1 3 1 11 17 GV/CV:R-
REG -1. 5 6 1 1 11 18 GV/CV:O+
REG -1. 5 6 3 3 1 11 19 GV/CV:O-
REG .190 5 6 1 2 1 1 20 GV/CV:Z+(AIR-IN)
REG .380 5 6 3 3 1 1 20 GV/CV:Z+(AIR-IN)
REG .190 5 6 1 2 11 21 GV/CV:Z-(AIR-OUT)
REG .380 5 6 3 3 11 21 GV/CV:Z-(AIR-OUT)
REG -1. 8 8 3 3 6 10 22 IHX.INT:R+
REG -1. 8 8 3 3 6 10 23 IHX.INT:R-
REG -1. 8 8 3 3 6 10 24 IHX.INT:O+
REG -1. 8 8 3 3 6 10 25 IHX.INT:O-
REG .4869 8 8 3 3 10 26 IHX.INT:Z+(NA-IN)
REG .4869 8 8 3 3 10 27 IHX.INT:Z-(NA-OUT)
REG -1. 2 2 1 3 3 9 28 RV.INTO:SEP.I=1&2:R+
REG -1. 1 1 3 3 9 29 RV.INTO:SEP.I=1&2:R-
REG -1. 1 1 1 4 6 30 RV.EXTO:ASY:O+
REG -1. 1 1 2 2 4 6 30 RV.INTO:ASY:O+
REG -1. 1 1 3 4 6 30 RV.INTO:ASY:O+
REG -1. 2 2 1 1 2 9 30 RV.EXTO:PIPE:O+
REG -1. 2 2 2 2 3 8 30 RV.INTO:PUMP/PIPE:O+
REG -1. 2 2 3 4 9 30 RV.INTO:PUMP/IHX:O+
REG -1. 1 1 1 4 6 31 RV.INTO:ASY:O-
REG -1. 1 1 2 2 4 6 31 RV.INTO:ASY:O-
REG -1. 1 1 3 4 6 31 RV.EXTO:ASY:O-
REG -1. 2 2 1 1 3 8 31 RV.INTO:PUMP/PIPE:O-
REG -1. 2 2 2 4 9 31 RV.INTO:PUMP/IHX:O-
REG -1. 2 2 3 7 9 31 RV.EXTO:IHX:O-
REG -1. 2 2 3 3 3 32 RV.INTO:BELOW.PUMP:IHX:Z+
REG -1. 2 2 1 2 10 32 RV.INTO:ABOVE.PIPE:PUMP:Z+
REG -1. 2 2 2 3 2 33 RV.INTO:BELOW.PUMP:IHX:Z-
REG -1. 2 2 1 2 9 33 RV.INTO:ABOVE.PIPE:PUMP:Z-
END
$DATA
IBOIL=0, IFENER=1, IFPROP=0,
ISETEN=1, ISETMO=1, ISTATE=0, NEWTS=1, NEWREB=0, NEWFOR=1,
ITIBUG=2, IENBUG=1, INOBUG=1,
TSTART=0., NTMAX=1000, TIMAX=1.0E+20, TREST=20.,

```

```

IDTIME=2, IHTWAL(99)=2,
IT=0.011.0, LASTDT=100, IT=2*1, ITMAXP=249, ITMAXE=249,
EPS1=1.0E-09, EPS2=1.0E-13, EPS3=1.0E-07, EPS5=1.0E-09,
NTHCON=-1.9*0,

KFLOW=-3.1,2*-3.1,-3, -6.1, 2*1,2*-3.2*1,-6, 2*1,2*-3.1,-5,
4*1,1,-5, 6*1,
VELOC=6*0., 2*0., 7*0., 4*0.-2.484,0.,
4*0.,1.32,0., 6*0.,
KTEMP=6*400, 400,400, 7*400, 4*400,1,400,
4*400,1,400, 6*400,
TEMP=4*395.,321.,468., 468.,395., 7*200., 4*70.,38.,100.,
4*395.,282.,395., 6*395.,
KPRES=6*0, 1,0, 6*0,1, 5*0,1,
5*0,1, 6*0,
PRES=6*0., 1.013E+5,0., 6*0.,1.013E+5, 5*0.,1.013E+5,
5*0.,1.013E+5, 6*0.,
KPROP=6*1, 2*1, 7*2, 6*3, 6*1, 6*1,
TVAL=0.,1.E10, 0.,1.E10,
FVAL=1.,1., 1.,1.,
NEND=2, 2, 0,

GRAVX=0., GRAVY=0., GRAVZ=-9.8,
TEMPO(1)=321.1, PRES0(1)=0.1013E+06,
XPRES0(1)=2.2, YPRES0(1)=0., ZPRES0(1)=13.18,
TEMPO(2)=200., PRES0(2)=0.1013E+06,
XPRES0(2)=2.964, YPRES0(2)=0., ZPRES0(2)=13.18,
GASW(2)=39.9, FCOR0(2)=0.,
FCOH(2)=1.4E+05, FC1H(2)=520.0,
FCOMU(2)=2.35E-05, FC1MU(2)=4.33E-08,
FCOK(2)=1.7E-02, FC1K(2)=4.0E-05,
TEMPO(3)=70., PRES0(3)=0.1013E+06,
XPRES0(3)=3.173, YPRES0(3)=0., ZPRES0(3)=12.15,
GASW(3)=28.8, FCOR0(3)=0.,
FCOH(3)=2.8E+05, FC1H(3)=1040.0,
FCOMU(3)=1.81E-05, FC1MU(3)=3.61E-08,
FCOK(3)=2.48E-02, FC1K(3)=5.73E-05,
FCTLO=250., FCTHI=750.,
NMATER=2,
COK= 4.63, 15.0,
C1K= 1.85E-02, 0.013,
C2K= 2.8E-06, 0.0,
CCCP= 210.0, 480.0,
C1CP= 0.0, 0.1736,
C2CP= 0.0, 0.0,
C0RC= 15400.0, 7979.0,
C1RC= 0.0, 0.0,
C2RC= 0.0, 0.0,

NTPLCT=-10.0,
NTPRNT=-9999,0,
ISTPR= -06201,06203, -10201,10203, -11201,11203, -12201,12203,
-01201,01203, -02201,02202, -03201,03203,
-05201,05203, -06201,06203, -07201,07203,
-15201,15203,
-60001,80005, 0,
:CHFE= -01201,01203, -02201,02202, -03201,03203,
-05201,05203, -06201,06203, -07201,07203,

```

```

-09201.09003, -15201.15203,
-80001.80005, 0,

NHEATC=7,
HEATC1= 5.0, 5.0, 5.0, 6.16, 1.0, 1.0,
5.0,
HEATC2= 0.401, 0.283, 0.025, 0.527, 1.22, 1.22,
0.025,
HEATC3= 0.45, 0.45, 0.8, 0.653, 0.457, 0.457,
0.8,
HEATC4= 0.45, 0.45, 0.8, 0.653, 0.4, 0.4,
0.8,

ICORR= 1,2,3, 4,5, 6,6,
FORCEF= 3*0.5, 2*0.5, 2*0.5,
REYLEN= 3*3.39E-03, .0171,.0190, .286,.078,
CLENGTH= 3*3.39E-03, .0171,.0190, .286,.078,
NCORR=6,
ACORRL= 0.,104.7,0., 2*51.9, 64.,
BCORRL= 3*-1., 2*-1., -1.,
CCORRL= .0663,0.,.0037, .0046,.386, 0.,
REYTRN= 3*1674., 2*1189., 1189.,
ACORRT= 0.,.400,0., 2*.256, .316,
BCORRT= 3*-.25, 2*-.25, -.25,
CCORRT= .0663,0.,.0037, .0046,.386, 0.,

IFRAD=2, NFRAD=4, OMR=0.05,
EMITS= 4*0.7,
FVIEW= 0., .96543, 0., 0., 95*0.,
1., .03457, 0., 0., 95*0.,
0., 0., 0., .90876, 95*0.,
0., 0., 1., .09124, 95*0.,
9405*0.,

IFPUMP=4,
TURBV=2.0E-3, TURBC=0.0,
&END
HTEX 2 3 4 1 3 1 11 RV/GV:ARGON
HTEX 2 4 4 1 1 12 12 RV/GV(EXP):ARGON
HTEX 3 5 6 1 3 1 11 GV/CV:AIR
END
&PUMP
IPUBUG=3, NPUMP=1, DENRO=8575.,
ICORP(1)=1, QFLOWR(1)=0.64368, PUMPHR(1)=84.7,
TORQR(1)=26981., NTORQM(1)=2,
PUMPS(1)=1116., PUMPSR=1116., PINER(1)=1182.,
QCUT=1.0E-09, CUTPF=0.01,.268,
CORPF=.01,0.,-.73.13, .00268,0.,.07, .00383,.01071,.01406,
&END
PUMZ 1 2 2 2 2 2 7 7 PRI.PUMP
END
&PUMP LOC.
ZFOR 1 1 1 1 3 3 3 DR: IN
ZFOR 2 1 1 1 3 4 5 DR: PIN
ZFOR 3 1 1 1 3 6 6 DR: OUT
ZFOR 4 2 2 3 3 6 9 IHX(PRI): TUBE
ZFOR 5 8 8 3 3 6 9 IHX(INT): SHELL
ZFOR 6 2 2 1 1 1 8 PIPE
ZFOR 7 2 2 2 2 2 6 PUMP.DUCT

```

```

FUEL PINS
&T N=1, IXYZ=3, NT=0, RODFR=1000., OUTR=9.525E-03, &END
&F IHT=1, HYD=3.39E-03, &END
&M MI=2, NP=1, DR=.457E-03, Q=0., SGAP=1.E+10, &END
&M MI=1, NP=5, DR=.555E-03, Q=539.04E+06, &END
IHX TUBES
&T N=2, IXYZ=3, NT=0, RODFR=1000., OUTR=9.525E-03, &END
&F IHT=4, HYD=19.0E-03, &END
&M MI=2, NP=3, DR=.326E-03, Q=0., &END
&F IHT=3, HYD=17.1E-03, &END
RV WALL
&T N=3, IXYZ=3, NT=0, RODFR=-.7251, OUTR=2.870, &END
&F IHT=5, HYD=0.2032, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=1, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=2, DR=2.539E-2, Q=0., KRAD=0, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=0, &END
&F IHT=7, HYD=0.0762, &END
GV WALL
&T N=4, IXYZ=3, NT=0, RODFR=-.3015, OUTR=3.023, &END
&F IHT=6, HYD=0.610, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=3, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=2, DR=2.539E-2, Q=0., KRAD=0, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=2, &END
&F IHT=5, HYD=0.2032, &END
CV WALL
&T N=5, IXYZ=3, NT=0, RODFR=-.2856, OUTR=3.3528, &END
&M MI=2, NP=1, DR=2.539E-2, Q=0., KRAD=0, SGAP=1.E-10, HGAP=1.E+10, &END
&M MI=2, NP=1, DR=1.E-5, Q=0., KRAD=4, &END
&F IHT=6, HYD=0.610, &END
END THERMAL STRUCT. PROTOTYPE
OUT 1 1 1 1 3 5 5 DR.PIN: FUEL (Q>0)
OUT 2 8 8 3 3 7 9 IHX.INT (SHELL)
IN 2 2 2 3 3 7 9 IHX.PRI (TUBE)
OUT 3 3 3 1 3 1 11 RV:ARGON
IN 3 2 2 1 3 1 11 RV:NA
OUT 4 5 5 1 3 1 11 GV:AIR-FLOW
IN 4 4 4 1 3 1 11 GV:ARGON
IN 5 6 6 1 3 1 11 CV:AIR-FLOW
END THERMAL STRUCT. LOC.
VELB -2.484 5 6 1 3 11 11 21 GV/CV:OUT
VELB -1.32 8 8 3 3 10 10 27 IHX.INT:OUT
END BNDRY. VALUE INIT.
AL .0627 1 1 1 3 4 6 ASY:DR
AL .0627 1 1 1 3 3 6 ASY:DR
AL .1022 2 2 3 3 7 9 IHX.PRI:TUBE
AL .1022 2 2 3 3 6 9 IHX.PRI:TUBE
AL .0376 2 2 2 2 7 8 PUMP:DUCT
AL .0376 2 2 2 2 6 8 PUMP:DUCT
AL .1143 2 2 1 1 3 8 PIPE
AL .1143 2 2 1 1 2 8 PIPE
AL .6712 3 4 1 3 1 11 RV/GV:ARGON
AL .6712 4 4 1 1 12 12 RV/GV:ARGON
AL .6724 3 4 1 2 1 11 RV/GV:ARGON
AL .6712 3 4 1 3 1 10 RV/GV:ARGON
AL .6712 4 4 1 1 11 11 RV/GV:ARGON
AL .889 5 6 1 3 1 11 GV:CV:AIR (FLOW)

```

ALY	.889	5	5	5	1	2	1	11	GV/CV:AIR(FLOW)
ALZ	.889	5	5	1	3	1	10	GV/CV:AIR(FLOW)	
AL	.213	5	8	3	3	6	10	IHX.INT:SHELL	
ALZ	.213	5	8	3	3	6	9	IHX.INT:SHELL	
P	1.0E+06	2	2	2	2	8	9	PUMP.OUT	
P	1.0E+06	2	2	1	1	3	9	PIPE	
P	1.0E+06	2	2	1	1	3	1	2	PIPE/CORE.IN
P	1.0E+06	1	1	1	3	1	4	CORE.IN	
TL	468.	1	1	1	3	5	11	ASY.EXIT/HOT.POOL	
TL	468.	2	2	1	3	10	11	HOT.POOL	
TL	419.	2	2	1	1	12	12	HOT.POOL(EXP.CELL)	
TL	370.	2	2	3	3	9	9	IHX.PRI	
TL	395.0	8	8	3	3	8	8	IHX.PRI	
WL	15.2	2	2	2	2	6	10	IHX.INT	
VL	-.27	2	2	1	1	9	9	PUMP/PIPE	
WL	-5.0	2	2	1	1	2	8	PIPE	
UL	-0.4	1	1	1	3	1	2	PIPE/CORE.IN	
WL	.34	1	1	1	3	2	2	CORE.IN	
WL	5.43	1	1	1	3	3	6	ASY:DR	
WL	.34	1	1	1	3	7	9	HOT.POOL	
UL	.07	1	1	1	3	10	11	HOT.POOL/IHX	
WL	-2.85	2	2	3	3	6	9	IHX.PRI:TUBE	
WL	-.29	2	2	3	3	3	5	IHX/PUMP	
VL	-.97	2	2	2	2	3	3	IHX/PUMP	
WL	.57	2	2	2	2	3	5	IHX/PUMP	
WL	.001	3	3	1	3	1	10	RV/GV	
WL	-.001	4	4	1	3	1	10	RV/GV	
WL	2.484	5	6	1	3	1	10	GV/CV	
WL	1.32	8	8	3	3	6	9	IHX.INT	
END								INTERNAL CELL INIT.	

THE FOLLOWING VARIABLES DETERMINE STORAGE ALLOCATION. IT IS IMPORTANT THAT THEY ARE ADEQUATELY SPECIFIED.

NM1	205	COMPUTATIONAL CELLS	IWIRE	0	WIRE WRAP OPTION FLAG
NL1	482	SURFACE ELEMENTS	ITURKE	0	TURBULENCE MODEL FLAG
NFORCE	7	FORCE STRUCTURES	ISYMC	5	SOLUTION TECHNIQUE FLAG
IMAX	8	CELLS IN THE X- OR P-DIRECTION	IFITEN	3	IMPLICIT ENERGY CALCULATION FLAG
JMAX	3	CELLS IN THE Y- OR THETA DIRECTION	IFREB	0	REBALANCING FLAG
KMAX	12	CELLS IN THE Z-DIRECTION	ISTRUC	1	THERMAL STRUCTURES FLAG
IFHTX	3	MULTI-FLUID FLAG	IGEOM	-1	GEOMETRY FLAG

64405 IS THE MINIMUM DIMENSION OF S NEEDED FOR THE ABOVE VALUES.
 80000 IS THE CURRENT DIMENSION OF S IN COMMON /SPACE/ IN SUBROUTINE ALTER.
 UNFORMATTED FILE CMXNEW OPENED ON UNIT 10

***** BOUNDARY SURFACE SUMMARY *****

BINARY STRING -- PRINTED CHARACTER TABLE

000000 --	000001 --	6	000010 --	5	000011 --	W	000100 --	4	000101 --	P	000110 --	T	000111 --	5
001000 --	001001 --	D	001010 --	H	001011 --	9	001100 --	V	001101 --	#	001110 --	S	001111 --	2
010000 --	010001 --	K	010010 --	L	010011 --	*	010100 --	R	010101 --	Q	010110 --	S	010111 --	&
011000 --	011001 --	E	011010 --	G	011011 --	<	011100 --	(	011101 --	:	011110 --	:	011111 --	?
100000 --	100001 --	J	100010 --	I	100011 --	*	100100 --	N	100101 --	Q	100110 --	M	100111 --	-
101000 --	101001 --	C	101010 --	A	101011 --	0	101100 --	=	101101 --	^	101110 --	@	101111 --	?
110000 --	110001 --	:	110010 --	+	110011 --	Y	110100 --	7	110101 --	:	110110 --	/	110111 --	?
111000 --	111001 --	.	111010 --	>	111011 --	?	111100 --	X	111101 --	?	111110 --	?	111111 --	?

***** J = 1 PLANE *****

K I-->

12	?	?
11	CFCFCE	
10	BGBFBF	
9	I-BFBF	
8	IXBFBF	
7	IXBFBF	
6	XXBFBF	
5	XXBFBF	
4	XXBFBF	
3	IXBFBF	
2	BFBFBF	
1	AGAGAG	

***** J = 2 PLANE *****

K I-->

12	JKJKJK
11	1L1212
10	U,1212
9	UX1212
8	UX1212
7	XX1212
6	XX1212
5	XX1212
4	XX1212
3	U>1212
2	1K1212
1	ILILIL

***** J = 3 PLANE *****

K I-->

12	QQQQQQ
11	NNNNR ?
10	7XNNR X
9	7XNNR X
8	7XNNR X
7	7XNNR X
6	XXNNR ?

5 XXNRNR
 4 XXNRNR
 3 7/NNNR
 2 NQNRNR
 1 MMSMS

CELLS AND SURFACE ELEMENTS HAVE BEEN COUNTED. CHECK THESE VALUES AGAINST YOUR EXPECTED TOTALS.

NM 0 IRREGULAR COMPUTATIONAL CELLS
 NM1 205 COMPUTATIONAL CELLS (TOTAL)
 NL 0 IRREGULAR SURFACE ELEMENTS
 NL1 482 SURFACE ELEMENTS (TOTAL)

TIME LEFT TO END OF RUN IS 2.95E+02 SECONDS.

***** GRID SUMMARY *****

	X	IMAX= 8	DX	Y	JMAX= 3	DY	Z	KMAX= 12	DZ	
1	7.750000E-01	1.550000E+00	1.9635000E-01	3.9270000E-01	7.500000E-02	1.500000E-01	1			1
2	2.200000E+00	1.300000E+00	5.890500E-01	3.927000E-01	2.250000E-01	1.500000E-01	2			2
3	2.888000E+00	7.600000E-02	1.178100E+00	7.854000E-01	5.550000E-01	5.100000E-01	3			3
4	2.964000E+00	7.600000E-02			1.440000E+00	1.260000E+00	4			4
5	3.087500E+00	1.710000E-01			2.935000E+00	1.730000E+00	5			5
6	3.258500E+00	1.710000E-01			4.510000E+00	1.420000E+00	6			6
7	3.424500E+00	1.610000E-01			6.110000E+00	1.780000E+00	7			7
8	3.880000E+00	7.500000E-01			7.890000E+00	1.780000E+00	8			8
9					9.670000E+00	1.780000E+00	9			9
10					1.109000E+01	1.060000E+00	10			10
11					1.215000E+01	1.060000E+00	11			11
12					1.318000E+01	1.000000E+00	12			12

***** SURFACE-SURFACE ELEMENT SUMMARY *****

SURFACE TYPE	SURFACE NUMBER	NUMBER OF SURFACE ELEMENTS	UNIT NORMAL VECTOR	
REGULAR	1	34	1.0000 I	0.0000 J
REGULAR	2	15	-1.0000 I	0.0000 J
REGULAR	3	12	0.0000 I	1.0000 J
REGULAR	4	17	0.0000 I	-1.0000 J
REGULAR	5	6	0.0000 I	0.0000 J
REGULAR	6	5	0.0000 I	0.0000 J
REGULAR	7	1	0.0000 I	0.0000 J
REGULAR	8	19	-1.0000 I	0.0000 J
REGULAR	9	34	1.0000 I	0.0000 J
REGULAR	10	34	-1.0000 I	0.0000 J
REGULAR	11	23	0.0000 I	1.0000 J
REGULAR	12	23	0.0000 I	-1.0000 J
REGULAR	13	6	0.0000 I	0.0000 J

```

REGULAR 14 0 0.0000 1 0.0000 J -1.0000 K
REGULAR 15 1 0.0000 I 0.0000 J -1.0000 K
REGULAR 16 33 1.0000 I 0.0000 J 0.0000 K
REGULAR 17 33 -1.0000 I 0.0000 J 0.0000 K
REGULAR 18 22 0.0000 I 1.0000 J 0.0000 K
REGULAR 19 22 0.0000 I -1.0000 J 0.0000 K
REGULAR 20 6 0.0000 I 0.0000 J 1.0000 K
REGULAR 21 6 0.0000 I 0.0000 J -1.0000 K
REGULAR 22 5 1.0000 I 0.0000 J 0.0000 K
REGULAR 23 5 -1.0000 I 0.0000 J 0.0000 K
REGULAR 24 5 0.0000 I 1.0000 J 0.0000 K
REGULAR 25 5 0.0000 I -1.0000 J 0.0000 K
REGULAR 26 1 0.0000 I 0.0000 J 1.0000 K
REGULAR 27 1 0.0000 I 0.0000 J -1.0000 K
REGULAR 28 21 1.0000 I 0.0000 J 0.0000 K
REGULAR 29 21 -1.0000 I 0.0000 J 0.0000 K
REGULAR 30 29 0.0000 I 1.0000 J 0.0000 K
REGULAR 31 24 0.0000 I -1.0000 J 0.0000 K
REGULAR 32 4 0.0000 I 0.0000 J 1.0000 K
REGULAR 33 4 0.0000 I 0.0000 J -1.0000 K

```

***** FLAG SUMMARY *****

```

IBOIL = 0 IDDDP = 1 IDRAG = 0 IDTIME= 2 IENBUG= 1 IFENER= 1
IFITEN= 3 IFPROP= 0 IFREB = 0 IFRES = 1 IFROD = 0 IGEOM = -1
IMAX = 8 IPART = 0 IREBIT= 50 ISETEN= 1 ISETMO= 1 ISTATE= 0
ISTRUC= 1 ISYNCH= 5 ITENMX= 1 ITBUG= 2 ITMAXE= 249 ITMAXP= 249
ITURKE= 0 IT(1) = 1 IT(2) = 1 IWIRE = 0 IXREB = 0 IYREB = 0
IZREB = 0 JMAX = 3 KMAX = 12 LASTDT= 100 LASTIT= 99999 LMPRNT= 0
NCCORR = 6 NEWFOR= 1 NEWREB= 0 NEWTS = 1 NFORCE= 7 NHEATC= 7
NL1 = 482 NMATER= 2 NM1 = 205 NREBRT= 0 NSURF = 33 NTMAX = 1000

```

***** AUTOMATIC TIME STEP SIZE CONTROL INVOKED (IDTIME= 2, AND ITAUTO=0) *****

IF THE MAXIMUM FRACTIONAL CHANGE IN CELL KINETIC ENERGY OR FLUID TEMPERATURE IS:
 (A) GREATER THAN 0.0100, THEN THE TIME STEP WILL BE REPEATED WITH ITS SIZE REDUCED BY A FACTOR OF 2.00
 UNTIL THIS TEST IS SATISFIED. IF SIZE DECREASES BELOW 1.000E-05 SECONDS, THEN THE CALCULATION WILL TERMINATE.
 (B) LESS THAN 0.0040 FOR 5 CONSECUTIVE STEPS, THEN THE NEXT STEP WILL BE TAKEN WITH ITS SIZE INCREASED
 BY A FACTOR OF 2.00, WITH AN UPPER BOUND OF 1.000E+05 SECONDS.

IF THIS IS A STEADY-STATE CALCULATION, THEN ITAUTO WILL BE TURNED ON ONLY WHEN THE AUTOMATIC
 TIME STEP SIZE CRITERIA HAVE BEEN MET DURING "MANUAL" CONTROL

***** SODIUM PROPERTIES SUMMARY (FLUID#1) *****

THE FOLLOWING VALUES WERE COMPUTED FROM THE PROPERTY ROUTINES WITH PRESSURE (PRES0(1)) = 1.01300E+05

TEMPERATURE	ENTHALPY	DENSITY	CONDUCTIVITY	VISCOSITY	SPECIFIC HEAT	TEMPERATURE
250.000	6.930E+05	8.918E+02	7.916E+01	3.923E-04	1.320E+03	250.000
300.000	7.586E+05	8.800E+02	7.658E+01	3.446E-04	1.304E+03	300.000
350.000	8.234E+05	8.681E+02	7.405E+01	3.080E-04	1.290E+03	350.000
400.000	8.876E+05	8.562E+02	7.159E+01	2.791E-04	1.278E+03	400.000
450.000	9.512E+05	8.442E+02	6.918E+01	2.557E-04	1.269E+03	450.000
500.000	1.014E+06	8.322E+02	6.683E+01	2.364E-04	1.262E+03	500.000

550.000	1.077E+06	8.202E+02	6.455E+01	2.203E-04	1.257E+03	550.000
600.000	1.143E+06	8.062E+02	6.232E+01	2.065E-04	1.254E+03	600.000
650.000	1.203E+06	7.961E+02	6.014E+01	1.947E-04	1.254E+03	650.000
700.000	1.266E+06	7.840E+02	5.803E+01	1.844E-04	1.257E+03	700.000
750.000	1.329E+06	7.719E+02	5.598E+01	1.754E-04	1.261E+03	750.000

***** SIMPLIFIED PROPERTIES SUMMARY FOR FLUID # 2 *****

PROPERTIES ARE COMPUTED WITH THE FOLLOWING FUNCTIONS:

ENTHALPY = 1.4000E+05 + TC * 5.2000E+02
 DENSITY = 3.9900E+01 * PRESSURE / (8314. * (TC + 273.16))
 CONDUCTIVITY = 1.7000E-02 + TC * 4.0000E-05
 VISCOSITY = 2.3500E-05 + TC * 4.3300E-08
 TEMPERATURE = -2.6923E+02 + H * 1.9231E-03

THE FOLLOWING VALUES WERE COMPUTED FROM THE ABOVE FUNCTIONS:

TEMPERATURE	ENTHALPY	DENSITY	CONDUCTIVITY	VISCOSITY	SPECIFIC HEAT	TEMPERATURE
250.000	2.700E+05	9.293E-01	2.700E-02	3.432E-05	5.200E+02	250.000
300.000	2.960E+05	8.482E-01	2.900E-02	3.649E-05	5.200E+02	300.000
350.000	3.220E+05	7.801E-01	3.100E-02	3.865E-05	5.200E+02	350.000
400.000	3.480E+05	7.222E-01	3.300E-02	4.082E-05	5.200E+02	400.000
450.000	3.740E+05	6.723E-01	3.500E-02	4.298E-05	5.200E+02	450.000
500.000	4.000E+05	6.288E-01	3.700E-02	4.515E-05	5.200E+02	500.000
550.000	4.260E+05	5.906E-01	3.900E-02	4.731E-05	5.200E+02	550.000
600.000	4.520E+05	5.568E-01	4.100E-02	4.948E-05	5.200E+02	600.000
650.000	4.780E+05	5.266E-01	4.300E-02	5.164E-05	5.200E+02	650.000
700.000	5.040E+05	4.996E-01	4.500E-02	5.381E-05	5.200E+02	700.000
750.000	5.300E+05	4.751E-01	4.700E-02	5.597E-05	5.200E+02	750.000

*** NOTE: THE ABOVE SAMPLE DENSITY VALUES WERE COMPUTED USING PRES0(2) = 1.013000E+05 IN THE IDEAL GAS
 LAW EQUATION; THE ACTUAL LOCAL PRESSURE WILL BE USED DURING THE REST OF THE CALCULATION.

***** SIMPLIFIED PROPERTIES SUMMARY FOR FLUID # 3 *****

PROPERTIES ARE COMPUTED WITH THE FOLLOWING FUNCTIONS:

ENTHALPY = 2.8000E+05 + TC * 1.0400E+03
 DENSITY = 2.8800E+01 * PRESSURE / (8314. * (TC + 273.16))
 CONDUCTIVITY = 2.4800E-02 + TC * 5.7300E-05
 VISCOSITY = 1.8100E-05 + TC * 3.6100E-08
 TEMPERATURE = -2.6923E+02 + H * 9.6154E-04

THE FOLLOWING VALUES WERE COMPUTED FROM THE ABOVE FUNCTIONS:

TEMPERATURE	ENTHALPY	DENSITY	CONDUCTIVITY	VISCOSITY	SPECIFIC HEAT	TEMPERATURE
250.000	5.400E+05	6.707E-01	3.912E-02	2.712E-05	1.040E+03	250.000
300.000	5.920E+05	6.122E-01	4.199E-02	2.893E-05	1.040E+03	300.000
350.000	6.440E+05	5.631E-01	4.485E-02	3.073E-05	1.040E+03	350.000
400.000	6.960E+05	5.213E-01	4.772E-02	3.254E-05	1.040E+03	400.000
450.000	7.480E+05	4.852E-01	5.058E-02	3.434E-05	1.040E+03	450.000
500.000	8.000E+05	4.539E-01	5.345E-02	3.615E-05	1.040E+03	500.000
550.000	8.520E+05	4.263E-01	5.631E-02	3.795E-05	1.040E+03	550.000
600.000	9.040E+05	4.019E-01	5.918E-02	3.976E-05	1.040E+03	600.000

550.000	9.560E+05	3.801E-01	5.204E-02	4.156E-05	1.040E+03	650.000
700.000	1.008E+06	3.606E-01	5.491E-02	4.337E-05	1.040E+03	700.000
750.000	1.060E+06	3.430E-01	6.777E-02	4.517E-05	1.040E+03	750.000

*** NOTE: THE ABOVE SAMPLE DENSITY VALUES WERE COMPUTED USING PRES0(3) = 1.01300E+05 IN THE IDEAL GAS LAW EQUATION; THE ACTUAL LOCAL PRESSURE WILL BE USED DURING THE REST OF THE CALCULATION.

***** MATERIAL PROPERTIES (OTHER THAN COOLANT) SUMMARY *****

NUMBER OF MATERIAL PROPERTIES SPECIFIED (NMATER) = 2
MATERIAL PROPERTIES ARE COMPUTED WITH THE FOLLOWING FUNCTIONS:

DENSITY = C0RG + C1RO * TC + C2RO * TC**2
SPECIFIC HEAT = C0CP + C1CP * TC + C2CP * TC**2
CONDUCTIVITY = C0K + C1K * TC + C2K * TC**2

MATERIAL	C0RO	DENSITY (KG/M**3)		C2RO	C0P	SPECIFIC HEAT (J/KG-C)		C2CP	THERMAL CONDUCTIVITY (WATT/M-C)		
		C1RO	C2RO			C1CP	C2CP		C0K	C1K	C2K
1	1.540E+04	0.000E+00	0.000E+00	0.000E+00	2.100E+02	0.000E+00	0.000E+00	0.000E+00	4.630E+00	1.850E-02	2.800E-06
2	7.979E+03	0.000E+00	0.000E+00	0.000E+00	4.800E+02	1.736E-01	0.000E+00	0.000E+00	1.500E+01	1.300E-02	0.000E+00

***** FLUID-STRUCTURE HEAT TRANSFER CORRELATIONS SUMMARY *****

NUMBER OF HEAT TRANSFER CORRELATIONS SPECIFIED (NHEATC) = 7
THE NUSSELT NUMBER IS COMPUTED WITH THE FOLLOWING FUNCTION:

NUSSELT = HEATC1 + HEATC2 * REYNOLDS NUMBER ** HEATC3 * PRANDTL NUMBER ** HEATC4

CORRELATION	HEATC1	HEATC2	HEATC3		HEATC4
			HEATC3	HEATC4	
1	5.000E+00	4.010E-01	4.500E-01	4.500E-01	4.500E-01
2	5.000E+00	2.830E-01	4.500E-01	4.500E-01	4.500E-01
3	5.000E+00	2.500E-02	8.000E-01	8.000E-01	8.000E-01
4	6.160E+00	5.270E-01	6.530E-01	6.530E-01	6.530E-01
5	1.000E+00	1.220E+00	4.570E-01	4.570E-01	4.570E-01
6	1.000E+00	1.220E+00	4.570E-01	4.570E-01	4.570E-01
7	5.000E+00	2.500E-02	8.000E-01	8.000E-01	8.000E-01

***** BOUNDARY CONDITIONS SUMMARY *****

SURFACE	KTEMP	KFLOW	KPRES	KPROP	MATWAL	IHTWAL	TEMP	VELOC	PRES	WALLDX	HYDWAL	WALLQS	TSINK	HSINK
1	400	-3	0	1	0	0	3.950E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	400	1	0	1	0	0	3.950E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	400	-3	0	1	0	0	3.950E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	400	-3	0	1	0	0	3.950E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5	400	1	0	1	0	0	3.210E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6	400	-3	0	1	0	0	4.680E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7	400	-6	1	1	0	0	4.680E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
8	400	1	0	1	0	0	3.950E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
9	400	1	0	2	0	0	2.000E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
10	400	1	0	2	0	0	2.000E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
11	400	-3	0	2	0	0	2.000E+02	0.000E+02	0.000E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00


```

2.400E+09 2.500E+09 2.600E+09 3.000E+09 3.200E+09 3.400E+09 3.600E+09 3.800E+09 4.000E+09 4.200E+09 4.400E+09 4.600E+09
4.800E+09 5.000E+09 5.200E+09 5.400E+09 5.600E+09 5.800E+09 6.000E+09 6.200E+09 6.400E+09 6.600E+09 6.800E+09 7.000E+09
7.200E+09 7.400E+09 7.600E+09 7.800E+09 8.000E+09 8.200E+09 8.400E+09 8.600E+09 8.800E+09 9.000E+09 9.200E+09 9.400E+09
9.600E+09 9.800E+09 1.000E+10

*** FUNCTION VALUES ***

1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00
1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00
1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00
1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00
1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00 1.000E+00

*** TOTAL NUMBER CELLS OF FLUID #1 = 72
*** TOTAL NUMBER CELLS OF FLUID #2 = 67
*** TOTAL NUMBER CELLS OF FLUID #3 = 66

***** PUMP MODEL SUMMARY *****
TOTAL NUMBER OF PUMPS IN THIS COMPUTATION = 1
NP PUMPSR QFLOWR PUMPHR TORQR PUMPS CUTPF(1) CUTPF(2) CUTPF(3) QCUT NCUT
1 1.11600E+03 6.43680E-01 8.47000E+01 2.69810E+04 1.11600E+03 1.00000E-02 2.68000E-01 1.00000E+00 1.00000E-09 1.00000E-03

```

```

*** IFPUMP= 4 ***, TOTAL NO. OF CORRELATIONS= 0
*** THE SUMMARY OF CORRELATION COEFFICIENTS ARE ***

```

NC	CORPH8(NC) (FULL PUMP OPERATIONAL RANGE)	CORPT8(NC) (IFPUMP=3, FULL PUMP OPERATIONAL RANGE)	CORPT8(NC) (IFPUMP=4, REVERSE SPEED OPERATION PRECLUDED)
1	6.986526053300E-01	-6.002091924720E-01	4.889663120000E+01
2	-8.494177047200E-01	1.398114216490E+00	-1.241188706000E+02
3	2.212716128110E+00	-1.554669068320E+00	1.214496110000E+02
4	-2.938053426030E+00	3.261332436010E+00	-5.219548958000E+01
5	2.025779620420E+00	-3.069693873450E+00	5.140216254000E+00
6	-7.151586008380E-01	1.349285702340E+00	3.780294746000E+00
7	1.306606055000E-01	-3.013840976240E-01	-1.506546075000E+00
8	-1.176382128290E-02	3.321642431680E-02	2.17097906000E-01
9	4.121832708230E-04	-1.433831779990E-03	-1.175690460000E-02

```

*** FRICTIONAL TORQUE = TORQR * ( C0 + C1 * NBAR + C2 * NBAR ** 2)

```

PUMP SPEED RANGE	C0	C1	C2	(C0 C2 C3)
NBAR > CUTPF(2)	3.830000E-03	1.071000E-02	1.406000E-02	(CORPF(7) CORPF(8) CORPF(9))
CUTPF(2) > NBAR > CUTPF(1)	2.680000E-03	0.000000E+00	7.000000E-02	(CORPF(4) CORPF(5) CORPF(6))
CUTPF(3) > NBAR	1.000000E-02	0.000000E+00	-7.313000E+01	(CORPF(1) CORPF(2) CORPF(3))

```

NO. OF PUMP IS 1 IN (IIJJKK) 20207 IDIR= 3

```

```

***** FORCE STRUCTURE SUMMARY *****

```


TOTAL NUMBER OF FORCE STRUCTURES (NFORCE) = 7

N	ICORR(N)	CLNTH(N)	REYLEN(N)	FORCEF(N)
1	1	3.3900E-03	3.3900E-03	5.0000E-01
2	2	3.3900E-03	3.3900E-03	5.0000E-01
3	3	3.3900E-03	3.3900E-03	5.0000E-01
4	4	1.7100E-02	1.7100E-02	5.0000E-01
5	5	1.9000E-02	1.9000E-02	5.0000E-01
6	6	2.8600E-01	2.8600E-01	5.0000E-01
7	6	7.8000E-02	7.8000E-02	5.0000E-01

TOTAL NUMBER OF CORRELATION TYPES (NCORR) = 6

NT	REYTRN(NT)	ACORRL(NT)	BCORRL(NT)	CCORRL(NT)	ACORRT(NT)	BCORRT(NT)	CCORRT(NT)
1	1.6740E+03	0.0000E+00	-1.0000E+00	6.6300E-02	0.0000E+00	-2.5000E-01	6.6300E-02
2	1.6740E+03	1.0470E+02	-1.0000E+00	0.0000E+00	4.0000E-01	-2.5000E-01	0.0000E+00
3	1.6740E+03	0.0000E+00	-1.0000E+00	3.7000E-03	0.0000E+00	-2.5000E-01	3.7000E-03
4	1.1890E+03	5.1900E+01	-1.0000E+00	4.6000E-03	2.5600E-01	-2.5000E-01	4.6000E-03
5	1.1890E+03	5.1900E+01	-1.0000E+00	3.8600E-01	2.5600E-01	-2.5000E-01	3.8600E-01
6	1.1890E+03	6.4000E+01	-1.0000E+00	0.0000E+00	3.1600E-01	-2.5000E-01	0.0000E+00

TOTAL NUMBER OF SURFACES WITH FORCES SPECIFIED = 31

***** THERMAL STRUCTURE SUMMARIES *****

THERMAL STRUCTURE PROTOTYPE NUMBER	NUMBER OF SURFACE CELLS		HEAT TRANSFER CORRELATION INDEX		HYDRAULIC DIAMETER		ROD FRACTION	GEOMETRY TYPE	HEAT SOURCE TRANSIENT FUNCTION NUMBER	
	OUTSIDE	INSIDE	OUTSIDE	INSIDE	OUTSIDE	INSIDE				
1	1	3	0	1	0	3.390E-03	0.000E+00	-2.496E+03	3	1
2	2	3	3	4	3	1.900E-02	1.710E-02	1.000E+03	3	0
3	3	33	33	5	7	2.032E-01	7.620E-02	-7.251E-01	3	0
4	4	33	33	6	5	6.100E-01	2.032E-01	-3.015E-01	3	0
5	5	0	33	0	6	0.000E+00	6.100E-01	-2.856E-01	3	0

EDIT FOR EACH TS IS 2 BLOCKS:

BLOCK 1: DETAILED GEOMETRY:

MATERIAL	TYPE	VOLUME	HEAT SOURCE	COEFFICIENT	PARTITION	SIZE	SURFACE AREA	VOLUME	TOTAL S. AREAAUT	TOTAL S. AREAIN
----------	------	--------	-------------	-------------	-----------	------	-----------------	--------	---------------------	--------------------

BLOCK 2: INVENTORY (AT TEMP0(1)= 3.21100E+02 C):

MATRL	VOLUME (M**3)	MASS (KG)	HEAT CAPACITY (J/C)
-------	------------------	--------------	------------------------

TYPE 1

FLUID			
MATERIAL	2	0.0000E+00	4.5700E-04
GAP			8.3000E-05
MATERIAL	1	5.3904E+06	2.3000E+05
MATERIAL	1	5.3904E+06	5.5500E-04
MATERIAL	1	5.3904E+06	5.5500E-04
MATERIAL	1	5.3904E+06	5.5500E-04

MATERIAL	1	5.3904E+08	5.5500E-04	1.7405E+01	7.2449E-03	
MATERIAL	1	5.3904E+08	5.5500E-04	8.7026E+00	2.4150E-03	
				0.0000E+00		0.0000E+00
	1	1.97084E-01	3.03509E+03			1.6968E+02
	2	7.22001E-02	5.76084E+02	6.37369E+05		
GAP		1.19658E-02		3.08633E+05		
TOTAL		2.81250E-01	3.61118E+03	9.46003E+05		
TYPE	2					
FLUID						
MATERIAL	2	0.0000E+00	3.2600E-04	5.9847E+01	1.9176E-02	
MATERIAL	2	0.0000E+00	3.2600E-04	5.7799E+01	1.8509E-02	
MATERIAL	2	0.0000E+00	3.2600E-04	5.5751E+01	1.7841E-02	
						3.1958E+02
FLUID				5.3702E+01		2.8577E+02
	2	2.96508E-01	2.36584E+03	1.26748E+06		
TOTAL		2.96508E-01	2.36584E+03	1.26748E+06		
TYPE	3					
FLUID						
MATERIAL	2	0.0000E+00	1.0000E-05	1.3076E+01	1.3076E-04	
GAP			1.0000E+10	1.3075E+01	1.3074E-09	
MATERIAL	2	0.0000E+00	2.5390E-02	1.3075E+01	3.3052E-01	
MATERIAL	2	0.0000E+00	2.5390E-02	1.2960E+01	3.2758E-01	
GAP			1.0000E+10	1.2844E+01	1.2843E-09	
MATERIAL	2	0.0000E+00	1.0000E-05	1.2844E+01	1.2844E-04	
						5.7162E+01
FLUID				1.2844E+01		5.6151E+01
	2	2.87815E+00	2.29648E+04	1.23032E+07		
GAP		1.13301E-08				
TOTAL		2.87815E+00	2.29648E+04	1.23032E+07		
TYPE	4					
FLUID						
MATERIAL	2	0.0000E+00	1.0000E-05	5.7267E+00	5.7267E-05	
GAP			1.0000E+10	5.7267E+00	5.7260E-10	
MATERIAL	2	0.0000E+00	2.5390E-02	5.7267E+00	1.4479E-01	
MATERIAL	2	0.0000E+00	2.5390E-02	5.6786E+00	1.4357E-01	
GAP			1.0000E+10	5.6305E+00	5.6297E-10	
MATERIAL	2	0.0000E+00	1.0000E-05	5.6305E+00	5.6305E-05	
						6.0221E+01
FLUID				5.6305E+00		5.9209E+01
	2	3.03353E+00	2.42045E+04	1.29674E+07		
GAP		1.19415E-08				
TOTAL		3.03353E+00	2.42045E+04	1.29674E+07		
TYPE	5					
MATERIAL	2	0.0000E+00	2.5390E-02	6.0165E+00	1.5218E-01	
GAP			1.0000E+10	5.9710E+00	5.9703E-10	
MATERIAL	2	0.0000E+00	1.0000E-05	5.9710E+00	5.9710E-05	
						6.6773E+01
FLUID				5.9709E+00		6.6267E+01
	2	1.68961E+00	1.34814E+04	7.22254E+06		
GAP		6.62599E-09				
TOTAL		1.68961E+00	1.34814E+04	7.22254E+06		

***** THERMAL STRUCTURE INVENTORY *****

** (MASS AND HEAT CAPACITY AT TEMP(1)= 3.21100E+02 C) **

MATRL	VOLUME (M**3)	MASS (KG)	HEAT CAPACITY (J/C)
1	1.97084E-01	3.03509E+03	6.37369E+05
2	7.96999E+00	6.35926E+04	3.40693E+07
GAP	1.19659E-02		
TOTAL	8.17904E+00	6.66277E+04	3.47066E+07

***** SUMMARY OF RADIATION HEAT TRANSFER INPUT *****

IFRAD = 2 RADIATION HEAT TRANSFER FLAG
 NRAD = 4 TOTAL NUMBER OF RADIATIVE SURFACES
 OMR = 0.0500 RELAXATION FACTOR FOR RADIATION HEAT TRANSFER

THERMAL STRUCTURE NO.	KRAD OF		KRAD OF OUTER SURFACE	INNER SURFACE		OUTER SURFACE AREA
	INNER SURFACE	OUTER SURFACE		AREA		
1	0	0	0	0.000000E+00	1.696832E+02	
2	0	0	0	2.867707E+02	3.195848E+02	
3	0	1	1	5.615062E+01	5.716242E+01	
4	2	3	3	5.920910E+01	6.022108E+01	
5	4	4	0	6.626696E+01	6.677282E+01	

RADIATING
SURFACE NO.
(KRAD)

RADIATING SURFACE NO. (KRAD)	EMISIVITY	THERMAL STRUCTURE NO.		SURFACE SIDE	SURFACE AREA
		3	4		
1	7.000000E-01	3		OUTER	5.716242E+01
2	7.000000E-01	4		INNER	5.920910E+01
3	7.000000E-01	4		OUTER	6.022108E+01
4	7.000000E-01	5		INNER	6.626696E+01

FRACTION OF HEAT RADIATED FROM SURFACE I RECEIVED BY SURFACE J

J	I-->			
	1	2	3	4
1	0.0000E+00	9.6543E-01	0.0000E+00	0.0000E+00
2	1.0000E+00	3.4570E-02	0.0000E+00	0.0000E+00
3	0.0000E+00	0.0000E+00	0.0000E+00	9.0876E-01
4	0.0000E+00	0.0000E+00	1.0000E+00	9.1240E-02

*** INITIAL DENSITY FOR FLUID #1 = 8.749809E+02
 *** INITIAL DENSITY FOR FLUID #2 = 1.027459E+00
 *** INITIAL DENSITY FOR FLUID #3 = 1.022575E+00

***** SURFACE AREA SUMMARY *****

SURFACE	AREA (M**2)
1	6.026950E-01
2	2.408508E-02
3	1.706950E-01

```

7 2.56650E+01
8 6.37941E+00
9 5.05639E+00
10 1.12312E+00
11 5.37939E+01
12 5.79146E+01
13 6.09719E+01
14 2.00336E+00
15 2.00336E+00
16 6.98616E-01
17 6.10155E-01
18 5.93750E-02
19 5.97930E+01
20 6.66042E+01
21 4.33656E+00
22 1.52000E+00
23 1.52000E+00
24 2.15271E+01
25 2.61334E+01
26 5.86500E+00
27 4.86900E-01
28 4.86900E-01
29 2.49804E+01
30 5.77385E+01
31 5.11475E+01
32 5.61561E+00
33 5.61561E+00

```

TOTAL SURFACE AREA = 6.04797E+02 M**2 TOTAL VOLUME = 8.01892E+01 M**3

***** TURBULENCE MODEL SUMMARY *****

ITURKE=	0	AKAPPA=	4.1992E-01	EPS6 =	1.0000E-09	PRNDLD=	1.3000E+00	CT1	=	1.4400E+00	TURBV =	2.0000E-03
ITMAXK=	99	CDTURB=	9.0000E-02	OMEGAD=	7.0000E-01	PRNDLK=	1.0000E+00	CT2	=	1.9200E+00	TURBC =	0.0000E+00
ITDBUG=	0	EE	= 9.0816E+00	OMEGAK=	7.0000E-01	PRNDLH=	9.0000E-01	TDIN	=	4.1000E-01	CHAPRE=	0.0000E+00
ITKBUG=	0	HYDIN	= 3.8800E+00	OMEGAT=	7.0000E-01			TKIN	=	1.0000E-03	CHART =	0.0000E+00
NTJUMP=	1			RELAXK=	8.0000E-01						TURBCO=	0.0000E+00

.....

GLOBAL BALANCES AT TIME= 0.000E+00 SECONDS

SURFACE NUMBER	MASS CONVECTED (KG/SEC)	ENERGY CONVECTED (WATT)	ENERGY CONDUCTED (WATT)	AVERAGE VELOCITY (M/SEC)	AVERAGE ENTHALPY (J/KG)	AVERAGE TEMPERATURE (CELSIUS)	BULK TEMPERATURE (CELSIUS)
N= 1	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.740418E+05	4.680000E+02	0.000000E+00
N= 2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	7.954319E+05	3.279262E+02	0.000000E+00
N= 3	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.581798E+05	4.555170E+02	0.000000E+00
N= 4	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.102535E+05	4.180731E+02	0.000000E+00
N= 5	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.870847E+05	3.211000E+02	0.000000E+00
N= 6	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.740501E+05	4.680000E+02	0.000000E+00
N= 7	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.740418E+05	4.680000E+02	0.000000E+00
N= 8	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.611744E+05	3.796536E+02	0.000000E+00
N= 9	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.440000E+05	2.000000E+02	0.000000E+00
N=10	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.440000E+05	2.000000E+02	0.000000E+00
N=11	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.440000E+05	2.000000E+02	0.000000E+00
N=12	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.440000E+05	2.000000E+02	0.000000E+00
N=13	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.440000E+05	2.000000E+02	0.000000E+00
N=14	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.440000E+05	2.000000E+02	0.000000E+00
N=15	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.440000E+05	2.000000E+02	0.000000E+00
N=16	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.528000E+05	7.000000E+01	0.000000E+00
N=17	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.528000E+05	7.000000E+01	0.000000E+00
N=18	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.528000E+05	7.000000E+01	0.000000E+00
N=19	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.528000E+05	7.000000E+01	0.000000E+00
N=20	4.263063E+00	1.362134E+06	-1.875559E+01	2.484000E+00	3.195200E+05	3.800000E+01	3.800000E+01
N=21	-3.849561E+00	-1.358125E+06	0.000000E+00	-2.476694E+00	3.528000E+05	7.000000E+01	7.000000E+01
N=22	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.812310E+05	3.950000E+02	0.000000E+00
N=23	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.812310E+05	3.950000E+02	0.000000E+00
N=24	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.812310E+05	3.950000E+02	0.000000E+00
N=25	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.812310E+05	3.950000E+02	0.000000E+00
N=26	5.683137E+02	4.177874E+08	-5.567434E+03	1.320000E+00	7.351352E+05	2.820000E+02	2.820000E+02
N=27	-5.509543E+02	-4.855028E+08	0.000000E+00	-1.319768E+00	8.812034E+05	3.950000E+02	3.950000E+02
N=28	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.027933E+05	3.338341E+02	0.000000E+00
N=29	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.418331E+05	4.426576E+02	0.000000E+00
N=30	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.417209E+05	3.642499E+02	0.000000E+00
N=31	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.488841E+05	3.698104E+02	0.000000E+00
N=32	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.612968E+05	3.798600E+02	0.000000E+00
N=33	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	7.870524E+05	3.211000E+02	0.000000E+00

***** MASS BALANCE *****

MASS ACCUMULATION RATE:

NET MASS CONVECTED INTO CONFIGURATION:

DMASDT= 0.000000E+00 (KG/SEC)

FLOSUM= 1.777287E+01 (KG/SEC)

GLOBAL CONSERVATION OF MASS:

(DMASDT-FLOSUM)=-1.777287E+01 (KG/SEC)

DLSUM= 0.000000E+00

***** FLUID ENERGY BALANCE *****

ENERGY ACCUMULATION RATE:

NET ENERGY CONVECTED INTO CONFIGURATION:

NET ENERGY CONDUCTED INTO CONFIGURATION:

ENERGY TRANSFERRED TO GENERATED WITHIN FLUID:

ENERGY EXCESS DUE TO MASS RESIDUE:

EDDT= 0.000000E+00 (WATT)

EINSUM=-6.771141E+07 (WATT)

QINSUM=-5.566189E+03 (WATT)

QSOURC= 1.062361E+08 (WATT)

EDLSUM= 0.000000E+00 (WATT)

GLOBAL CONSERVATION OF ENERGY: (DEDT-EINSUM-QINSUM-QSCUPC-EDLSUM)=-3.851911E+07 (WATT)

***** FLUID INVENTORY *****

FLUID	MASS (KG)	ENERGY (J)	HL (J/KG)	TL (C)
1	4.713573E+04	4.1398873934E+10	8.782907E+05	3.927364E+02
2	6.174218E+00	1.5065091216E+06	2.440000E+05	2.000000E+02
3	2.965997E+01	6.9360379353E+06	3.528000E+05	7.000000E+01
		4.1407316481E+10		

***** THERMAL STRUCTURE HEAT BALANCE *****

STRUCTURE NUMBER	IN SOLID (W)	SIDE=OUT	SIDE=IN	NET	IN SOLID (W)	NET HEAT SOURCE	STORED ENERGY	TEMPERATURE
							.GT.TEMP0(I) (J)	AVG (C)
1	1.062361E+08	1.062361E+08	0.000000E+00	1.062361E+08	-3.576279E-05	2.3736997381E+08	5.678579E+02	
2	0.000000E+00	-9.267814E+07	9.267814E+07	-3.337860E-06	3.337860E-06	8.2676353997E+07	3.850077E+01	
3	0.000000E+00	3.661026E+03	-3.661026E+03	4.418835E-07	-4.418835E-07	4.4520101070E+08	3.568828E+02	
4	0.000000E+00	3.052664E+03	-3.052664E+03	3.115565E-08	-3.115565E-08	-2.9268884072E+09	7.590785E+01	
5	0.000000E+00	0.000000E+00	1.945090E-08	1.945090E-08	-1.945090E-08	-1.6660179317E+09	7.000000E+01	
TOTAL	1.062361E+08			1.062361E+08	-3.242493E-05	-3.8276590004E+09	2.060495E+02	

***** RADIATIVE SURFACES *****

SURFACE NUMBER	TEMPERATURE (CELSIUS)	HEAT TRANSFER TO SURF (WATT)
1	3.638632E+02	0.000000E+00
2	7.599536E+01	0.000000E+00
3	7.582225E+01	0.000000E+00
4	7.000000E+01	0.000000E+00

==> Deleted rest (13 pages, as selected by ISTPR) of t=0 long edit. <==

***** CONTROL SUMMARY *****

DDHMX	0.0000E+00	DLCT	5.0000E-01	DTIME	1.0000E-02	AO	5.0000E-01	CDRAGX	1.0000E+00
DTENER	1.0000E-02	EPS1	1.0000E-09	DTFUEL	1.0000E+40	BO	0.0000E+00	CDRAGY	1.0000E+00
EPS5	1.0000E-09	EPS2	1.0000E-13	DTWALL	1.0000E+40	GRAVY	0.0000E+00	CDRAGZ	1.0000E+00
ITENMX	1	EPS3	1.0000E-07	DT(1)	1.0000E-02	GRAVZ	0.0000E+00		
ITMAXE	249	IREBIT	50	DT(2)	1.0000E+00	GRAVZ	-9.8000E+00		
OMEGAE	9.0000E-01	ITERMX	1	LASTDT	100	ALPHA	1.0000E+00		
OMEGAR	1.0000E+00	ITMAXP	249	RDTIME	8.0000E-01				
RELAXE	9.5000E-01			TMAX	1.0000E+20			CWIREX	5.0000E-01
ITAUTO	0	IT(1)	1	TIMEUP	2.95E+02			CWIREY	5.0000E-01
CHGMIN	4.0000E-03	IT(2)	1	TREST	2.00E+01			CWIREZ	5.0000E-01
CHGMAX	1.0000E-02	LASTIT	99999	TSTART	0.0000E+00			HYDALL	0.0000E+00
DTFACT	2.0000E+00	OMEGA	1.5000E+00					HYDIN	3.8800E+00
DT(3)	1.0000E-05	OMEGAV	1.0000E+00					HYDOUT	0.0000E+00
DT(4)	1.0000E+05								
1: TEMP0	3.2110E+02	PRES0	1.0130E+05	XPRES0	2.2000E+00	YPRES0	0.0000E+00	ZPRES0	1.1180E+01
2: TEMP0	2.0000E+02	PRES0	1.0130E+05	XPRES0	2.9640E+00	YPRES0	0.0000E+00	ZPRES0	1.3180E+01
3: TEMP0	7.0000E+01	PRES0	1.0130E+05	XPRES0	3.1730E+00	YPRES0	0.0000E+00	ZPRES0	1.2150E+01

```

NT      TIME      DTIME      IT      MAX(DLV) / DCONV      MAX(DU) / VELMAX      MAX(DV) / VELMAX      MAX(DW) / VELMAX      TLEFT      COURDT      DCONV      VELMAX
NP= 1  PUMPH= 8.68263E+01 PUMPHR= 8.47000E+01 TORQ=-2.67328E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.75044E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.4453533E+05
PSSR= 1.00000E+00 QFSR= 9.97214E-01 XPUMPH= 3.92560E+00 HTOT= 5.13982E-01 XTORT= 4.96782E-01
C.G.: ITR= 41 DLMAX= 3.70294E-06 DCONV= 8.57824E-06 FACTORIZED? YES CPU= 0.38E-03 SEC.
... MOLOOP ITMO= 1 DU=-7.079E-02 AT 10102 DV= 6.392E-02 AT 20102 ITP= 41 DLMAX= 3.703E-06 AT 30302
... ENLOOP... DHOHT=-1.348E-02 IJKT= 50301 HL= 3.480E+05 DHOH=-1.348E-02 IJK= 50301 HL= 3.480E+05
N= 1 TROD= 3.64224E+02 QRN=-1.321625E+04 QROLD= 0.00000E+00 AREATR= 5.716242E+01 OMR= 5.00000E-02
N= 2 TROD= 7.561663E+01 QRN=-1.321621E+04 QROLD= 0.00000E+00 AREATR= 5.920910E+01 OMR= 5.00000E-02
N= 3 TROD= 7.545554E+01 QRN=-8.570245E+01 QROLD= 0.00000E+00 AREATR= 6.022108E+01 OMR= 5.00000E-02
N= 4 TROD= 6.994095E+01 QRN= 8.570199E+01 QROLD= 0.00000E+00 AREATR= 6.626696E+01 OMR= 5.00000E-02
* 1 1.0000E-02 1.00E-02 1 4.32E-01 -7.08E-02 6.39E-02 1.93E-02 -1.35E-02 295.43 1.02E-01 8.58E-06 1.52E+01

NP= 1  PUMPH= 8.70925E+01 PUMPHR= 8.47000E+01 TORQ=-2.66873E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.74589E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.4681812E+05
PSSR= 1.00000E+00 QFSR= 9.89249E-01 XPUMPH= 3.92159E+00 HTOT= 5.19680E-01 XTORT= 4.99902E-01
C.G.: ITR= 33 DLMAX= 8.19039E-06 DCONV= 8.50953E-06 FACTORIZED? NO CPU= 0.12E-03 SEC.
... MOLOOP ITMO= 1 DU=-1.191E-03 AT 10101 DV= 1.779E-03 AT 20101 DW= 1.228E-02 AT 20208 ITP= 33 DLMAX= 8.190E-06 AT 30102
... ENLOOP... DHOHT=-1.149E-02 IJKT= 50301 HL= 3.440E+05 DHOH=-1.149E-02 IJK= 50301 HL= 3.440E+05
N= 1 TROD= 3.638582E+02 QRN=-2.535153E+04 QROLD=-1.321625E+04 AREATR= 5.716242E+01 OMR= 5.00000E-02
N= 2 TROD= 9.905758E+01 QRN= 2.535145E+04 QROLD= 1.321621E+04 AREATR= 5.920910E+01 OMR= 5.00000E-02
N= 3 TROD= 9.822425E+01 QRN=-5.652600E+02 QROLD=-8.570245E+01 AREATR= 6.022108E+01 OMR= 5.00000E-02
N= 4 TROD= 7.002576E+01 QRN= 5.652571E+02 QROLD= 8.570199E+01 AREATR= 6.626696E+01 OMR= 5.00000E-02
* 2 2.0000E-02 1.00E-02 1 9.62E-01 -1.19E-03 1.78E-03 1.23E-02 -1.15E-02 295.43 1.03E-01 8.51E-06 1.51E+01

NP= 1  PUMPH= 8.66862E+01 PUMPHR= 8.47000E+01 TORQ=-2.67559E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.75275E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.4333383E+05
PSSR= 1.00000E+00 QFSR= 1.00136E+00 XPUMPH= 3.92767E+00 HTOT= 5.11030E-01 XTORT= 4.95155E-01
C.G.: ITR= 30 DLMAX= -7.74894E-06 DCONV= 8.61386E-06 FACTORIZED? NO CPU= 0.19E-03 SEC.
... MOLOOP ITMO= 1 DU=-9.914E-04 AT 10101 DV= 1.651E-03 AT 20101 DW= 1.043E-02 AT 20207 ITP= 30 DLMAX= 7.749E-06 AT 50302
... ENLOOP... DHOHT=-9.894E-04 IJKT= 50301 HL= 3.406E+05 DHOH=-9.894E-04 IJK= 50301 HL= 3.406E+05
N= 1 TROD= 3.635212E+02 QRN=-3.643710E+04 QROLD=-2.535153E+04 AREATR= 5.716242E+01 OMR= 5.00000E-02
N= 2 TROD= 1.199548E+02 QRN= 3.643698E+04 QROLD= 2.535145E+04 AREATR= 5.920910E+01 OMR= 5.00000E-02
N= 3 TROD= 1.185297E+02 QRN=-1.433519E+03 QROLD=-5.652600E+02 AREATR= 6.022108E+01 OMR= 5.00000E-02
N= 4 TROD= 7.076408E+01 QRN= 1.433511E+03 QROLD= 5.652571E+02 AREATR= 6.626696E+01 OMR= 5.00000E-02
* 3 3.0000E-02 1.00E-02 1 -9.00E-01 -9.91E-04 1.65E-03 1.04E-02 -9.89E-03 295.43 1.02E-01 8.61E-06 1.53E+01

NP= 1  PUMPH= 8.63283E+01 PUMPHR= 8.47000E+01 TORQ=-2.68122E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.75838E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.4026498E+05
PSSR= 1.00000E+00 QFSR= 1.01180E+00 XPUMPH= 3.93286E+00 HTOT= 5.03632E-01 XTORT= 4.91041E-01
C.G.: ITR= 30 DLMAX= 3.98507E-06 DCONV= 8.70394E-06 FACTORIZED? YES CPU= 0.19E-03 SEC.
... MOLOOP ITMO= 1 DU=-8.196E-04 AT 10101 DV= 1.536E-03 AT 20101 DW= 8.883E-03 AT 20207 ITP= 30 DLMAX= 3.985E-06 AT 60201
... ENLOOP... DHOHT=-8.505E-04 IJKT= 50301 HL= 3.377E+05 DHOH=-8.505E-04 IJK= 50301 HL= 3.377E+05
N= 1 TROD= 3.632277E+02 QRN=-4.652226E+04 QROLD=-3.643710E+04 AREATR= 5.716242E+01 OMR= 5.00000E-02
N= 2 TROD= 1.382612E+02 QRN= 4.652211E+04 QROLD= 3.643698E+04 AREATR= 5.920910E+01 OMR= 5.00000E-02
N= 3 TROD= 1.363124E+02 QRN=-2.666678E+03 QROLD=-1.433519E+03 AREATR= 6.022108E+01 OMR= 5.00000E-02
N= 4 TROD= 7.214712E+01 QRN= 2.666664E+03 QROLD= 1.433511E+03 AREATR= 6.626696E+01 OMR= 5.00000E-02
* 4 4.0000E-02 1.00E-02 1 4.58E-01 -8.20E-04 1.54E-03 8.88E-03 -8.50E-03 294.63 1.01E-01 8.70E-06 1.54E+01

NP= 1  PUMPH= 6.60148E+01 PUMPHR= 8.47000E+01 TORQ=-2.68585E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.76302E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.3755683E+05
PSSR= 1.00000E+00 QFSR= 1.6203E+00 XPUMPH= 3.93728E+00 HTOT= 4.97314E-01 XTORT= 4.87489E-01
C.G.: ITR= 27 DLMAX= 7.43194E-06 DCONV= 8.78125E-06 FACTORIZED? NO CPU= 0.10E-03 SEC.
... MOLOOP ITMO= 1 DU=-7.439E-04 AT 10102 DV= 1.437E-03 AT 20101 DW= 7.561E-03 AT 20207 ITP= 27 DLMAX= 7.439E-06 AT 50102
... ENLOOP... DHOHT=-7.317E-03 IJKT= 50101 HL= 3.355E+05 DHOH=-7.317E-03 IJK= 50101 HL= 3.355E+05

```

```

N= 1 TROD= 3.62375E+02 QRN=-5.566815E+04 QROLD=-4.652220E+04 AREATR= 5.16142E+01 OMR= 5.000000E-02
N= 2 TROD= 1.541369E+02 QRN= 5.566797E+04 QROLD= 4.652211E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 1.551755E+02 QRN=-4.227647E+03 QROLD= 2.666678E+03 AREATR= 5.022118E+01 OMR= 5.000000E-02
N= 4 TROD= 7.413572E+01 QRN= 4.227624E+03 QROLD= 2.666664E+03 AREATR= 5.626696E+01 OMR= 5.000000E-02
# 5 5.00000E-02 1.00E-02 1 5.45E-01 7.50E-04 1.44E-03 7.55E-03 -7.33E-03 294.43 9.96E-02 1.59E+01

NP= 1 PUMPH= 8.57414E+01 PUMPHR= 8.47000E+01 TORQ=-2.68968E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.76685E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.3523290E+05
PSSR= 1.00000E+00 QFSR= 1.02851E+00 XPUMP= 3.94105E+00 HTOT= 4.91923E+01 XTORT= 4.84431E-01
C.G.: ITR= 26 DLMAX= -7.53066E-06 DCONV= 8.64765E-06 FACTORIZED? NO CPU= 0.23E-03 SEC
... MOLOOP ITMO= 1 DU= 7.985E-04 AT 10102 DV= 1.348E-03 AT 20101 DM= 6.437E-03 AT 20207 ITP= 26 DLMAX= 4.364E-06 AT 30102
... ENLOOP...DHQHT=6.313E-03 IJKT= 60301 HL= 3.336E+05 HLT= 3.358E+05 DHQH=6.313E-03 IJK= 60301 HL= 3.336E+05
N= 1 TROD= 3.627611E+02 QRN=-6.394498E+04 QROLD=-5.566815E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 1.677662E+02 QRN= 6.394478E+04 QROLD= 5.566797E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 1.649405E+02 QRN=-6.069165E+03 QROLD=-4.227647E+03 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 7.566892E+01 QRN= 6.069133E+03 QROLD= 4.227624E+03 AREATR= 6.626696E+01 OMR= 5.000000E-02
# 6 6.00000E-02 1.00E-02 1 -8.51E-01 7.99E-04 1.35E-03 6.44E-03 -6.31E-03 294.44 9.69E-02 1.57E+01

NP= 1 PUMPH= 8.55039E+01 PUMPHR= 8.47000E+01 TORQ=-2.69285E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.77001E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.3319611E+05
PSSR= 1.00000E+00 QFSR= 1.03513E+00 XPUMP= 3.94425E+00 HTOT= 4.87324E+01 XTORT= 4.81803E-01
C.G.: ITR= 28 DLMAX= -3.91065E-06 DCONV= 8.90461E-06 FACTORIZED? YES CPU= 0.27E-01 SEC
... MOLOOP ITMO= 1 DU= 8.325E-04 AT 10102 DV= 1.269E-03 AT 20101 DM= 5.481E-03 AT 20207 ITP= 28 DLMAX= 3.911E-06 AT 50202
... ENLOOP...DHQHT=5.489E-03 IJKT= 60301 HL= 3.318E+05 HLT= 3.336E+05 DHQH=5.489E-03 IJK= 60301 HL= 3.318E+05
N= 1 TROD= 3.625819E+02 QRN=-7.142829E+04 QROLD=6.394498E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 1.793524E+02 QRN= 7.142806E+04 QROLD= 6.394478E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 1.761590E+02 QRN=-8.137920E+03 QROLD=6.069165E+03 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 7.966872E+01 QRN= 8.137878E+03 QROLD= 6.069133E+03 AREATR= 6.626696E+01 OMR= 5.000000E-02
# 7 7.00000E-02 1.00E-02 1 -4.39E-01 8.33E-04 1.27E-03 5.48E-03 -5.49E-03 294.33 9.83E-02 1.58E+01

NP= 1 PUMPH= 8.52981E+01 PUMPHR= 8.47000E+01 TORQ=-2.69548E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.77265E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.3143135E+05
PSSR= 1.00000E+00 QFSR= 1.04081E+00 XPUMP= 3.94698E+00 HTOT= 4.83403E+01 XTORT= 4.79547E-01
C.G.: ITR= 25 DLMAX= 6.92053E-06 DCONV= 8.95342E-06 FACTORIZED? NO CPU= 0.12E-03 SEC
... MOLOOP ITMO= 1 DU= 8.545E-04 AT 10102 DV= 1.198E-03 AT 20101 DM= 4.667E-03 AT 20207 ITP= 25 DLMAX= 6.921E-06 AT 30202
... ENLOOP...DHQHT=5.050E-03 IJKT= 50202 HL= 3.409E+05 HLT= 3.427E+05 DHQH=5.050E-03 IJK= 50202 HL= 3.409E+05
**TIMSTP: AUTO TIME STEPS INITIATED *****
N= 1 TROD= 3.624341E+02 QRN=-7.819500E+04 QROLD=-7.142829E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 1.891247E+02 QRN= 7.819475E+04 QROLD= 7.142806E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 1.856025E+02 QRN=-1.037876E+04 QROLD=-8.137920E+03 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 8.304712E+01 QRN= 1.037878E+04 QROLD= 8.137878E+03 AREATR= 6.626696E+01 OMR= 5.000000E-02
# 9 8.00000E-02 1.00E-02 1 7.73E-01 8.54E-04 1.20E-03 4.67E-03 -5.05E-03 294.23 9.77E-02 1.59E+01

NP= 1 PUMPH= 8.51202E+01 PUMPHR= 8.47000E+01 TORQ=-2.69767E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.77484E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2990612E+05
PSSR= 1.00000E+00 QFSR= 1.04566E+00 XPUMP= 3.94931E+00 HTOT= 4.80059E+01 XTORT= 4.77613E-01
C.G.: ITR= 26 DLMAX= 4.36383E-06 DCONV= 8.99520E-06 FACTORIZED? NO CPU= 0.19E-03 SEC
... MOLOOP ITMO= 1 DU= 8.667E-04 AT 10102 DV= 1.135E-03 AT 20101 DM= 3.975E-03 AT 20207 ITP= 26 DLMAX= 4.364E-06 AT 30102
... ENLOOP...DHQHT=4.888E-03 IJKT= 50102 HL= 3.393E+05 HLT= 3.409E+05 DHQH=4.888E-03 IJK= 50102 HL= 3.393E+05
N= 1 TROD= 3.623142E+02 QRN=-8.432012E+04 QROLD=-7.819500E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 1.973170E+02 QRN= 8.431985E+04 QROLD= 7.819475E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 1.934995E+02 QRN=-1.273824E+04 QROLD=-1.037876E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 8.671299E+01 QRN= 1.273817E+04 QROLD= 1.037870E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
# 9 9.00000E-02 1.00E-02 1 4.85E-01 8.67E-04 1.13E-03 3.97E-03 -4.89E-03 294.10 9.73E-02 1.59E+01

NP= 1 PUMPH= 8.49668E+01 PUMPHR= 8.47000E+01 TORQ=-2.69950E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.77667E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2859069E+05
PSSR= 1.00000E+00 QFSR= 1.04982E+00 XPUMP= 3.95129E+00 HTOT= 4.77209E+01 XTORT= 4.75358E-01

```


C.G.: ITR= 26 DLMAX= 7.30863E-06 DCONV= 9.03095E-06 FACTORIZED? YES CPU= 0.12E-03 SEC.
 ... MLOOP ITWO= 1 DU= 8.709E-04 AT 10102 DV= 1.077E-03 AT 20101 DW= 3.385E-03 AT 20207 ITP= 26 DLMAX= 7.309E-06 AT 30232
 ... ENLOOP... DHOHT=-4.676E-03 IJKT= 50302 HL= 3.577E+05 DHOH=-4.676E-03 IJK= 50302 HL= 3.377E+05
 N= 1 TROD= 3.622187E+02 QRN=-8.987424E+04 QROLD=-8.432012E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
 N= 2 TROD= 2.041563E+02 QRN= 8.987396E+04 QROLD= 8.431985E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
 N= 3 TROD= 2.000727E+02 QRN=-1.516725E+04 QROLD=-1.273824E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
 N= 4 TROD= 9.057799E+01 QRN= 1.516718E+04 QROLD= 1.273817E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
 * 10 1.0000E-01 1.00E-02 1 8.09E-01 8.71E-04 1.08E-03 3.38E-03 -4.68E-03 294.01 9.69E-02 9.03E-06 1.60E+01
 UNFORMATTED FILE PLTDTAT OPENED ON UNIT 76

PLOTTING INFORMATION WRITTEN TO UNIT 76.

NP= 1 PUMPH= 8.48348E+01 PUMPHR= 8.47000E+01 TORQ=-2.70104E+04 TORQR= 2.69810E+04 TORQE=-7.71657E+05
 TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.77820E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.274562E+05
 PSSR= 1.00000E+00 QFSR= 1.05337E+00 XPUMPH= 3.95298E+00 HTOT= 4.74779E-01 XTORTQ= 4.74541E-01
 C.G.: ITR= 25 DLMAX= 7.10702E-06 DCONV= 9.06152E-06 FACTORIZED? NO CPU= 0.19E-03 SEC.
 ... MLOOP ITWO= 1 DU= 8.687E-04 AT 10102 DV= 1.025E-03 AT 20101 DW= 2.883E-03 AT 20207 ITP= 25 DLMAX= 7.107E-06 AT 30102
 ... ENLOOP... DHOHT=-4.434E-03 IJKT= 60102 HL= 3.366E+05 DHOH=-4.434E-03 IJK= 60102 HL= 3.366E+05
 N= 1 TROD= 3.621444E+02 QRN=-9.492199E+04 QROLD=-8.987424E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
 N= 2 TROD= 2.098545E+02 QRN= 9.492169E+04 QROLD= 8.987396E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
 N= 3 TROD= 2.05298E+02 QRN=-1.762266E+04 QROLD=-1.516725E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
 N= 4 TROD= 9.456089E+01 QRN= 1.762257E+04 QROLD= 1.516718E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
 * 11 1.0000E-01 1.00E-02 1 7.84E-01 8.69E-04 1.02E-03 2.88E-03 -4.43E-03 293.87 9.66E-02 9.06E-06 1.61E+01

NP= 1 PUMPH= 8.47213E+01 PUMPHR= 8.47000E+01 TORQ=-2.70323E+04 TORQR= 2.69810E+04 TORQE=-7.71657E+05
 TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.77949E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2648492E+05
 PSSR= 1.00000E+00 QFSR= 1.05641E+00 XPUMPH= 3.95441E+00 HTOT= 4.72708E-01 XTORTQ= 4.73329E-01
 C.G.: ITR= 27 DLMAX= -4.77069E-06 DCONV= 9.08764E-06 FACTORIZED? NO CPU= 0.29E-01 SEC.
 ... MLOOP ITWO= 1 DU= 8.614E-04 AT 10102 DV= 9.770E-04 AT 20101 DW= 2.455E-03 AT 20207 ITP= 27 DLMAX=-4.771E-06 AT 50302
 ... ENLOOP... DHOHT=-4.195E-03 IJKT= 60202 HL= 3.352E+05 DHOH=-4.195E-03 IJK= 60202 HL= 3.352E+05
 N= 1 TROD= 3.620884E+02 QRN=-9.952127E+04 QROLD=-9.492199E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
 N= 2 TROD= 2.146021E+02 QRN= 9.952096E+04 QROLD= 9.492169E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
 N= 3 TROD= 2.100581E+02 QRN=-2.006803E+04 QROLD=-1.762266E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
 N= 4 TROD= 9.859025E+01 QRN= 2.006792E+04 QROLD= 1.762257E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
 * 12 1.2000E-01 1.00E-02 1 -5.25E-01 8.61E-04 9.77E-04 2.46E-03 -4.19E-03 293.80 9.63E-02 9.09E-06 1.61E+01

NP= 1 PUMPH= 8.46238E+01 PUMPHR= 8.47000E+01 TORQ=-2.70340E+04 TORQR= 2.69810E+04 TORQE=-7.71657E+05
 TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78057E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2564939E+05
 PSSR= 1.00000E+00 QFSR= 1.05900E+00 XPUMPH= 3.95564E+00 HTOT= 4.70943E-01 XTORTQ= 4.72294E-01
 C.G.: ITR= 27 DLMAX= 5.99053E-06 DCONV= 9.10996E-06 FACTORIZED? YES CPU= 0.23E-03 SEC.
 ... MLOOP ITWO= 1 DU= 8.500E-04 AT 10102 DV= 9.332E-04 AT 20101 DW= 2.091E-03 AT 20207 ITP= 27 DLMAX= 5.991E-06 AT 30102
 ... ENLOOP... DHOHT=-3.943E-03 IJKT= 60102 HL= 3.352E+05 DHOH=-3.943E-03 IJK= 60102 HL= 3.352E+05
 N= 1 TROD= 3.620482E+02 QRN=-1.037231E+05 QROLD=-9.952127E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
 N= 2 TROD= 2.185659E+02 QRN= 1.037228E+05 QROLD= 9.952096E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
 N= 3 TROD= 2.138217E+02 QRN=-2.247376E+04 QROLD=-2.006803E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
 N= 4 TROD= 1.026058E+02 QRN= 2.247364E+04 QROLD= 2.006792E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
 * 13 1.3000E-01 1.00E-02 1 6.58E-01 8.50E-04 9.33E-04 2.09E-03 -3.94E-03 293.68 9.60E-02 9.11E-06 1.61E+01

NP= 1 PUMPH= 8.45403E+01 PUMPHR= 8.47000E+01 TORQ=-2.70432E+04 TORQR= 2.69810E+04 TORQE=-7.71657E+05
 TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78148E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2493297E+05
 PSSR= 1.00000E+00 QFSR= 1.06122E+00 XPUMPH= 3.95568E+00 HTOT= 4.69440E-01 XTORTQ= 4.71410E-01
 C.G.: ITR= 27 DLMAX= 6.61097E-06 DCONV= 9.12901E-06 FACTORIZED? NO CPU= 0.12E-03 SEC.
 ... MLOOP ITWO= 1 DU= 8.355E-04 AT 10102 DV= 8.928E-04 AT 20101 DW= 1.781E-03 AT 20207 ITP= 27 DLMAX= 6.611E-06 AT 30102
 ... ENLOOP... DHOHT=-3.686E-03 IJKT= 60102 HL= 3.352E+05 DHOH=-3.686E-03 IJK= 60102 HL= 3.352E+05
 N= 1 TROD= 3.620213E+02 QRN=-1.075022E+05 QROLD=-9.952127E+04 AREATR= 5.716242E+01 OMR= 5.000000E-02
 N= 2 TROD= 2.216658E+02 QRN= 1.075017E+05 QROLD= 9.952096E+04 AREATR= 5.920910E+01 OMR= 5.000000E-02
 N= 3 TROD= 2.169641E+02 QRN=-2.451622E+04 QROLD=-2.247376E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
 N= 4 TROD= 1.026058E+02 QRN= 2.451610E+04 QROLD= 2.006792E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02

```

N= 4 TRCD= 1.06583E+02 QRN= 1.43165E+04 QROLD= 2.24736E+04 AREATR= 6.62669E+01 OMR= 5.00000E-02
# 14 1.4000E-01 1.00E-02 1 7.14E-01 8.36E-04 8.93E-04 1.78E-03 -3.69E-03 293.46 9.57E-02 3.15E-06 1.62E+01

NP= 1 PUMPH= 8.44687E+01 PUMPHR= 8.47000E+01 TORQ=-2.70508E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQRF= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78225E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2379404E+05
PSSR= 1.00000E+00 QFSR= 1.06311E+00 XPUMPH= 3.95952E+00 HTOT= 4.68158E-01 XTORT= 4.70455E-01
C.G.: ITR= 27 DLMAX= 6.05520E-06 DCONV= 9.14527E-06 FACTORIZED? NO CPU= 0.13E-03 SEC.
... MOLOOP ITWO= 1 DU= 8.186E-04 AT 10102 DV= 8.554E-04 AT 20101 DW= 1.517E-03 AT 20207 ITP= 27 DLMAX= 6.055E-06 AT 30102
... ENLOOP... DHOHT=-3.431E-03 IJKT= 60202 HL= 3.315E+05 HLT= 3.326E+05 DHOH=-3.431E-03 IJK= 60202 HL= 3.315E+05
N= 1 TROD= 3.620059E+02 QRN=-1.111066E+05 QROLD=-1.075721E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.246897E+02 QRN= 1.111062E+05 QROLD= 1.075717E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.195939E+02 QRN=-2.707944E+04 QROLD=-2.481669E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.104094E+02 QRN= 2.707930E+04 QROLD= 2.481656E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
# 15 1.5000E-01 1.00E-02 1 6.62E-01 8.19E-04 8.55E-04 1.52E-03 -3.43E-03 293.46 9.57E-02 3.15E-06 1.62E+01

NP= 1 PUMPH= 8.44075E+01 PUMPHR= 8.47000E+01 TORQ=-2.70573E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQRF= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78290E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2379404E+05
PSSR= 1.00000E+00 QFSR= 1.06472E+00 XPUMPH= 3.95833E+00 HTOT= 4.67066E-01 XTORT= 4.70010E-01
C.G.: ITR= 27 DLMAX= 7.06377E-06 DCONV= 9.15915E-06 FACTORIZED? YES CPU= 0.43E-01 SEC.
... MOLOOP ITWO= 1 DU= 7.998E-04 AT 10102 DV= 8.207E-04 AT 20101 DW= 1.292E-03 AT 20207 ITP= 27 DLMAX= 7.064E-06 AT 30202
... ENLOOP... DHOHT=-3.180E-03 IJKT= 60202 HL= 3.304E+05 HLT= 3.315E+05 DHOH=-3.180E-03 IJK= 60202 HL= 3.304E+05
N= 1 TROD= 3.620002E+02 QRN=-1.143598E+05 QROLD=-1.111066E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.270684E+02 QRN= 1.143595E+05 QROLD= 1.111062E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.218176E+02 QRN=-2.924964E+04 QROLD=-2.707944E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.141303E+02 QRN= 2.924948E+04 QROLD= 2.707930E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
# 16 1.6000E-01 1.00E-02 1 7.71E-01 8.00E-04 8.21E-04 1.29E-03 -3.18E-03 293.35 9.55E-02 3.15E-06 1.62E+01

NP= 1 PUMPH= 8.43551E+01 PUMPHR= 8.47000E+01 TORQ=-2.70628E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQRF= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78345E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2334481E+05
PSSR= 1.00000E+00 QFSR= 1.06610E+00 XPUMPH= 3.95897E+00 HTOT= 4.66136E-01 XTORT= 4.69461E-01
C.G.: ITR= 27 DLMAX= 6.37777E-06 DCONV= 9.117098E-06 FACTORIZED? NO CPU= 0.19E-03 SEC.
... MOLOOP ITWO= 1 DU= 7.98E-04 AT 10102 DV= 7.884E-04 AT 20101 DW= 1.101E-03 AT 20207 ITP= 27 DLMAX= 6.378E-06 AT 30102
... ENLOOP... DHOHT=-2.937E-03 IJKT= 60202 HL= 3.294E+05 HLT= 3.304E+05 DHOH=-2.937E-03 IJK= 60202 HL= 3.294E+05
N= 1 TROD= 3.620030E+02 QRN=-1.173604E+05 QROLD=-1.143598E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.291054E+02 QRN= 1.173600E+05 QROLD= 1.143595E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.237115E+02 QRN=-3.131911E+04 QROLD=-2.924964E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.177002E+02 QRN= 3.131895E+04 QROLD= 2.924948E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
# 17 1.7000E-01 1.00E-02 1 6.95E-01 7.80E-04 7.88E-04 1.10E-03 -2.94E-03 293.28 9.54E-02 3.15E-06 1.62E+01

NP= 1 PUMPH= 8.43103E+01 PUMPHR= 8.47000E+01 TORQ=-2.70675E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQRF= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78391E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2296083E+05
PSSR= 1.00000E+00 QFSR= 1.06727E+00 XPUMPH= 3.95952E+00 HTOT= 4.65343E-01 XTORT= 4.68992E-01
C.G.: ITR= 27 DLMAX= 6.96678E-06 DCONV= 9.18108E-06 FACTORIZED? NO CPU= 0.34E-01 SEC.
... MOLOOP ITWO= 1 DU= 7.588E-04 AT 10102 DV= 7.581E-04 AT 20101 DW= 9.374E-04 AT 20207 ITP= 27 DLMAX= 6.967E-06 AT 30202
... ENLOOP... DHOHT=-2.705E-03 IJKT= 60202 HL= 3.286E+05 HLT= 3.294E+05 DHOH=-2.705E-03 IJK= 60202 HL= 3.286E+05
N= 1 TROD= 3.620128E+02 QRN=-1.201329E+05 QROLD=-1.173604E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.308658E+02 QRN= 1.201325E+05 QROLD= 1.173600E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.253397E+02 QRN=-3.328312E+04 QROLD=-3.131911E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.211054E+02 QRN= 3.328295E+04 QROLD= 3.131895E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
# 18 1.8000E-01 1.00E-02 1 7.59E-01 7.59E-04 7.58E-04 9.37E-04 -2.70E-03 293.14 9.53E-02 3.15E-06 1.63E+01
**AMAIN: DTL AND DEK HAVE BEEN LESS THAN CHGMIN FOR 5 STEPS; INCREASE THE TIME STEP SIZE TO 2.00000E-02 .
CONTROL CHANGE AT NTIME= 19 DTIME=2.000E-02 DTFUEL=1.000E+40 DTWALL=1.000E+40 DTENER=1.000E-02 ITERMX= 1

NP= 1 PUMPH= 8.42721E+01 PUMPHR= 8.47000E+01 TORQ=-2.70714E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQRF= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78430E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2263283E+05
PSSR= 1.00000E+00 QFSR= 1.06827E+00 XPUMPH= 3.95999E+00 HTOT= 4.64668E-01 XTORT= 4.68592E-01
C.G.: ITR= 31 DLMAX= 3.59832E-06 DCONV= 9.18968E-06 FACTORIZED? YES CPU= 0.35E-03 SEC.
... MOLOOP ITWO= 1 DU= 1.441E-03 AT 10102 DV= 1.423E-03 AT 20101 DW= 1.455E-03 AT 20207 ITP= 31 DLMAX= 3.598E-06 AT 30202

```

```

..ENLCOOP...DHOHT=-4.518E-03 IJKT= 60302 HL= 3.271E+05 HLT= 3.286E+05 DHOH=-4.518E-03 IJK= 60302 HL= 3.271E+05
N= 1 TROD= 3.621624E+02 QRN=-1.227108E+05 QROLD=-1.201329E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.324018E+02 QRN=-1.227104E+05 QROLD= 1.201325E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.267536E+02 QRN=-3.514025E+04 QROLD=-3.328312E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.243131E+02 QRN= 3.514007E+04 QROLD= 3.328295E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 19 2.0000E-01 2.00E-02 1 3.92E-01 1.44E-03 1.42E-03 1.46E-03 -4.52E-03 293.12 9.52E-02 9.19E-06 1.63E+01

NP= 1 PUMPH= 8.42125E+01 PUMPHR= 8.47000E+01 TORQ=-2.70775E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQER= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78491E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2212207E+05
PSSR= 1.00000E+00 QFSR= 1.06983E+00 XPUMPH= 3.96071E+00 HTOT= 4.63620E-01 XTORT= 4.67970E-01
C.G.: ITR= 31 DLMAX= 4.44825E-06 DCONV= 9.20306E-06 FACTORIZED? NO CPU= 0.19E-03 SEC.
...MOLCOOP ITWO= 1 DU= 1.355E-03 AT 10102 DV= 1.323E-03 AT 20101 DW= 1.056E-03 AT 20207 ITP= 31 DLMAX= 4.448E-06 AT 30302
..ENLCOOP...DHOHT=-4.065E-03 IJKT= 60203 HL= 3.417E+05 HLT= 3.431E+05 DHOH=-4.065E-03 IJK= 60203 HL= 3.417E+05
N= 1 TROD= 3.623053E+02 QRN=-1.251097E+05 QROLD=-1.227108E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.337821E+02 QRN= 1.251093E+05 QROLD= 1.227104E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.280203E+02 QRN=-3.689173E+04 QROLD=-3.514025E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.273502E+02 QRN= 3.689154E+04 QROLD= 3.514007E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 20 2.2000E-01 2.00E-02 1 4.83E-01 1.35E-03 1.32E-03 1.06E-03 -4.07E-03 293.12 9.51E-02 9.20E-06 1.63E+01

NP= 1 PUMPH= 8.41691E+01 PUMPHR= 8.47000E+01 TORQ=-2.70818E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQER= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78335E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.2175003E+05
PSSR= 1.00000E+00 QFSR= 1.07095E+00 XPUMPH= 3.96124E+00 HTOT= 4.62859E-01 XTORT= 4.67519E-01
C.G.: ITR= 23 DLMAX= -5.07757E-06 DCONV= 9.21278E-06 FACTORIZED? NO CPU= 0.19E-03 SEC.
...MOLCOOP ITWO= 1 DU= 1.270E-03 AT 10102 DV= 1.233E-03 AT 20101 DW= 7.804E-04 AT 60103 ITP= 23 DLMAX=-5.078E-06 AT 10102
..ENLCOOP...DHOHT=-3.969E-03 IJKT= 60203 HL= 3.404E+05 HLT= 3.417E+05 DHOH=-3.969E-03 IJK= 60203 HL= 3.404E+05
N= 1 TROD= 3.624412E+02 QRN=-1.273438E+05 QROLD=-1.251097E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.350227E+02 QRN= 1.273434E+05 QROLD= 1.251093E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.291552E+02 QRN=-3.853959E+04 QROLD= 3.689173E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.302158E+02 QRN= 3.853939E+04 QROLD= 3.689154E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 21 2.4000E-01 2.00E-02 1 -5.51E-01 1.27E-03 1.23E-03 -7.80E-04 -3.97E-03 293.12 9.50E-02 9.21E-06 1.63E+01

==== Deleted short edits (68 pages) for time step numbers 22 through 312. <==

NP= 1 PUMPH= 8.37103E+01 PUMPHR= 8.47000E+01 TORQ=-2.71257E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQER= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78974E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.1781590E+05
PSSR= 1.00000E+00 QFSR= 1.08277E+00 XPUMPH= 3.96671E+00 HTOT= 4.54945E-01 XTORT= 4.62793E-01
C.G.: ITR= 4 DLMAX= -9.00433E-06 DCONV= 9.21410E-06 FACTORIZED? NO CPU= 0.12E-03 SEC.
...MOLCOOP ITWO= 1 DU= 1.092E-08 AT 50103 DV=-2.679E-08 AT 60208 DW=-1.367E-07 AT 50207 ITP= 4 DLMAX=-9.004E-06 AT 10203
..ENLCOOP...DHOHT= 3.539E-07 IJKT= 40111 HL= 3.416E+05 HLT= 3.416E+05 DHOH= 3.539E-07 IJK= 40111 HL= 3.416E+05
N= 1 TROD= 3.918480E+02 QRN=-1.836551E+05 QROLD=-1.836551E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.763462E+02 QRN= 1.836545E+05 QROLD= 1.836545E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.676990E+02 QRN=-7.549524E+04 QROLD= 7.549523E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.883770E+02 QRN= 7.549485E+04 QROLD= 7.549484E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 313 3.4671E+06 1.00E+05 1 -9.77E-01 1.09E-08 -2.68E-08 -1.37E-07 3.54E-07 250.75 9.40E-02 9.21E-06 1.65E+01

NP= 1 PUMPH= 8.37103E+01 PUMPHR= 8.47000E+01 TORQ=-2.71257E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQER= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78974E+04 DENRO= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.1781590E+05
PSSR= 1.00000E+00 QFSR= 1.08277E+00 XPUMPH= 3.96671E+00 HTOT= 4.54945E-01 XTORT= 4.62793E-01
C.G.: ITR= 10 DLMAX= 3.56148E-06 DCONV= 9.21410E-06 FACTORIZED? NO CPU= 0.12E-03 SEC.
...MOLCOOP ITWO= 1 DU=-1.940E-08 AT 50102 DV= 2.966E-08 AT 60207 DW= 1.241E-07 AT 50206 ITP= 10 DLMAX= 3.561E-06 AT 10102
..ENLCOOP...DHOHT= 1.837E-07 IJKT= 30110 HL= 3.620E+05 HLT= 3.620E+05 DHOH= 1.837E-07 IJK= 30110 HL= 3.620E+05
N= 1 TROD= 3.918480E+02 QRN=-1.836551E+05 QROLD=-1.836551E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.763462E+02 QRN= 1.836546E+05 QROLD= 1.836545E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.676990E+02 QRN=-7.549524E+04 QROLD= 7.549523E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.883770E+02 QRN= 7.549485E+04 QROLD= 7.549484E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 314 3.5671E+06 1.00E+05 1 3.87E-01 -1.94E-09 2.97E-08 1.24E-07 1.84E-07 250.75 9.40E-02 9.21E-06 1.65E+01

```

```

NP= 1 PUMPH= 8.37103E+01 PUMPHR= 8.47000E+01 TORQ=-2.71257E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78974E+04 TORQR= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.1781590E+05
PSSR= 1.00000E+00 QFSR= 1.08277E+00 XPUMPH= 3.96671E+00 HTOT= 4.54945E-01 XTORT= 4.62793E-01
C.G.: ITR= 10 DLMAX= 6.81994E-06 DCONV= 9.21410E-06 FACTORIZED? YES CPU= 0.12E-03 SEC.
... MOLOOP ITMO= 1 DU=-2.278E-08 AT 50203 DV=-1.666E-08 AT 60210 DW= 9.895E-08 AT 50207 ITP= 10 DLMAX= 6.820E-06 AT 20302
... ENLOOP...DHOHT=-2.574E-07 IJKT= 30110 HL= 3.620E+05 HLT= 3.620E+05 DHOH=-2.574E-07 IJK= 30110 HL= 3.620E+05
N= 1 TROD= 3.918480E+02 QRN=-1.836551E+05 QROLD=-1.836551E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.763462E+02 QRN=-1.836546E+05 QROLD=-1.836546E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.676990E+02 QRN=-7.549526E+04 QROLD=-7.549526E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.883770E+02 QRN= 7.549487E+04 QROLD= 7.549486E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 315 3.6671E+06 1.00E+05 1 7.40E-01 -2.28E-08 -1.67E-08 9.89E-08 -2.57E-07 250.75 9.40E-02 9.21E-06 1.65E+01

NP= 1 PUMPH= 8.37103E+01 PUMPHR= 8.47000E+01 TORQ=-2.71257E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78974E+04 TORQR= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.1781590E+05
PSSR= 1.00000E+00 QFSR= 1.08277E+00 XPUMPH= 3.96671E+00 HTOT= 4.54945E-01 XTORT= 4.62793E-01
C.G.: ITR= 10 DLMAX= -5.26149E-06 DCONV= 9.21410E-06 FACTORIZED? NO CPU= 0.22E-03 SEC.
... MOLOOP ITMO= 1 DU= 1.446E-08 AT 50202 DV= 2.052E-08 AT 60209 DW= 9.938E-08 AT 50208 ITP= 10 DLMAX= 5.261E-06 AT 20103
... ENLOOP...DHOHT=-1.525E-07 IJKT= 40111 HL= 3.416E+05 HLT= 3.416E+05 DHOH=-1.525E-07 IJK= 40111 HL= 3.416E+05
N= 1 TROD= 3.918480E+02 QRN=-1.836552E+05 QROLD=-1.836551E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.763462E+02 QRN= 1.836546E+05 QROLD= 1.836546E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.676990E+02 QRN=-7.549527E+04 QROLD=-7.549526E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.883771E+02 QRN= 7.549487E+04 QROLD= 7.549487E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 316 3.671E+06 1.00E+05 1 -5.71E-01 1.45E-08 2.05E-08 9.94E-08 -1.52E-07 250.75 9.40E-02 9.21E-06 1.65E+01

NP= 1 PUMPH= 8.37103E+01 PUMPHR= 8.47000E+01 TORQ=-2.71257E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78974E+04 TORQR= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.1781590E+05
PSSR= 1.00000E+00 QFSR= 1.08277E+00 XPUMPH= 3.96671E+00 HTOT= 4.54945E-01 XTORT= 4.62793E-01
C.G.: ITR= 5 DLMAX= 8.32525E-06 DCONV= 9.21410E-06 FACTORIZED? NO CPU= 0.12E-03 SEC.
... MOLOOP ITMO= 1 DU= 1.010E-08 AT 50203 DV= 1.895E-08 AT 60210 DW= 9.253E-08 AT 50209 ITP= 5 DLMAX= 8.325E-06 AT 10201
... ENLOOP...DHOHT=-1.675E-07 IJKT= 30110 HL= 3.620E+05 HLT= 3.620E+05 DHOH=-1.675E-07 IJK= 30110 HL= 3.620E+05
N= 1 TROD= 3.918480E+02 QRN=-1.836552E+05 QROLD=-1.836552E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.763462E+02 QRN= 1.836546E+05 QROLD= 1.836546E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.676990E+02 QRN=-7.549528E+04 QROLD=-7.549527E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.883771E+02 QRN= 7.549488E+04 QROLD= 7.549487E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 317 3.6671E+06 1.00E+05 1 9.04E-01 1.01E-08 1.90E-08 9.25E-08 -1.67E-07 250.74 9.40E-02 9.21E-06 1.65E+01

NP= 1 PUMPH= 8.37103E+01 PUMPHR= 8.47000E+01 TORQ=-2.71257E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78974E+04 TORQR= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.1781590E+05
PSSR= 1.00000E+00 QFSR= 1.08277E+00 XPUMPH= 3.96671E+00 HTOT= 4.54945E-01 XTORT= 4.62793E-01
C.G.: ITR= 4 DLMAX= 7.85453E-06 DCONV= 9.21410E-06 FACTORIZED? YES CPU= 0.19E-03 SEC.
... MOLOOP ITMO= 1 DU=-1.077E-08 AT 50210 DV= 1.796E-08 AT 60211 DW=-8.159E-08 AT 50109 ITP= 4 DLMAX= 7.855E-06 AT 20103
... ENLOOP...DHOHT=-1.654E-07 IJKT= 40111 HL= 3.416E+05 HLT= 3.416E+05 DHOH=-1.654E-07 IJK= 40111 HL= 3.416E+05
N= 1 TROD= 3.918480E+02 QRN=-1.836552E+05 QROLD=-1.836552E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.763462E+02 QRN= 1.836546E+05 QROLD= 1.836546E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.676990E+02 QRN=-7.549529E+04 QROLD=-7.549528E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02
N= 4 TROD= 1.883771E+02 QRN= 7.549489E+04 QROLD= 7.549488E+04 AREATR= 6.626696E+01 OMR= 5.000000E-02
* 318 3.6671E+06 1.00E+05 1 8.52E-01 -1.08E-08 -1.80E-08 -8.16E-08 -1.65E-07 249.94 9.40E-02 9.21E-06 1.65E+01

NP= 1 PUMPH= 8.37103E+01 PUMPHR= 8.47000E+01 TORQ=-2.71257E+04 TORQR= 2.69810E+04 TORQF=-7.71657E+02
TORQFR= 0.00000E+00 TORQMT= 0.00000E+00 TORQM= 2.78974E+04 TORQR= 8.57500E+03 QCUT= 1.00000E-09 DPPUMP= 7.1781590E+05
PSSR= 1.00000E+00 QFSR= 1.08277E+00 XPUMPH= 3.96671E+00 HTOT= 4.54945E-01 XTORT= 4.62793E-01
C.G.: ITR= 4 DLMAX= 7.86283E-06 DCONV= 9.21410E-06 FACTORIZED? NO CPU= 0.12E-03 SEC.
... MOLOOP ITMO= 1 DU=-1.017E-08 AT 50211 DV=-1.121E-08 AT 60210 DW=-8.351E-08 AT 50210 ITP= 4 DLMAX= 7.883E-06 AT 20301
... ENLOOP...DHOHT=-9.416E-08 IJKT= 30110 HL= 3.620E+05 HLT= 3.620E+05 DHOH=-9.416E-08 IJK= 30110 HL= 3.620E+05
N= 1 TROD= 3.918480E+02 QRN=-1.836552E+05 QROLD=-1.836552E+05 AREATR= 5.716242E+01 OMR= 5.000000E-02
N= 2 TROD= 2.763462E+02 QRN= 1.836546E+05 QROLD= 1.836546E+05 AREATR= 5.920910E+01 OMR= 5.000000E-02
N= 3 TROD= 2.676990E+02 QRN=-7.549529E+04 QROLD=-7.549529E+04 AREATR= 6.022108E+01 OMR= 5.000000E-02

```

N= 4 TROP= 1.883771E+02 ORN= 7.549490E+04 QFOLD= 7.549489E+04 AREATR= 6.626696E+01 OMR= 5.000000E+02
 # 319 4.6671E+06 1.00E+05 1 8.56E-01 1.02E-08 -1.12E-08 -8.35E-08 -9.42E-08 249.93 9.40E 02 9.21E-06 1.45E+01

..... STEADY STATE HAS BEEN REACHED TO WITHIN EPS3= 1.000E-07
 #
 #

***** TIMING SUMMARY *****
 TOTAL CPU TIMSTEP EXECUTION CPU TIME = 4.379255E+01
 TOTAL MOMENTUM CONSTRUCTION CPU TIME = 7.620295E+00
 TOTAL MOMENTUM EQNS SOLUTION CPU TIME = 9.087274E+00
 TOTAL ENERGY EQ CONSTRUCTION CPU TIME = 1.281329E+01
 TOTAL ENERGY EQNS SOLUTION CPU TIME = 1.119402E+01
 TOTAL THERM STRUCT EXECUTION CPU TIME = 1.725340E+00
 TOTAL TS RADIATION EXECUTION CPU TIME = 2.570111E-01
 TOTAL TURB-KE EQ CONSTRUCT CPU TIME = 0.000000E+00
 TOTAL TURB-KE EQ SOLUTION CPU TIME = 0.000000E+00
 TOTAL TURB-KE-D EQ CONSTRUCT CPU TIME = 0.000000E+00
 TOTAL TURB-KE-D EQ SOLUTION CPU TIME = 0.000000E+00

RESTART FILE HAS BEEN WRITTEN AT TIME 4.067E+06 AFTER ITERATION 0 OF TIME STEP 319
 BLOCK LENGTHS OF OTHER COMMON BLOCKS (WRITTEN TO RESTART): 109 114 3409 13640 601 141 81 208 64405
 BLOCK LENGTHS OF COMMON /SPACE/ (WRITTEN TO RESTART) 12634 233 0 0 0 894 50643

PLOTTING INFORMATION APPENDED TO UNIT 10.

GLOBAL BALANCES AT TIME= 4.067E+06 SECONDS

SURFACE NUMBER	MASS CONVECTED (KG/SEC)	ENERGY CONVECTED (WATT)	ENERGY CONDUCTED (WATT)	AVERAGE VELOCITY (M/SEC)	AVERAGE ENTHALPY (J/KG)	AVERAGE TEMPERATURE (CELSIUS)	BULK TEMPERATURE (CELSIUS)
N= 1	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	1.013314E+06	4.990784E+02	0.000000E+00
N= 2	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.454415E+05	3.668509E+02	0.000000E+00
N= 3	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.983816E+05	4.872818E+02	0.000000E+00
N= 4	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.534620E+05	4.519911E+02	0.000000E+00
N= 5	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.371970E+05	3.601636E+02	0.000000E+00
N= 6	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	1.013293E+06	4.990555E+02	0.000000E+00
N= 7	-1.257179E-06	-1.284050E+00	0.000000E+00	-1.355320E-09	1.013314E+06	4.990784E+02	0.000000E+00
N= 8	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.979085E+05	4.161467E+02	0.000000E+00
N= 9	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.269823E+05	3.595814E+02	0.000000E+00
N=10	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.982792E+05	3.943831E+02	0.000000E+00
N=11	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.111459E+05	3.291368E+02	0.000000E+00
N=12	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.149619E+05	3.364552E+02	0.000000E+00
N=13	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	2.804471E+05	2.700906E+02	0.000000E+00
N=14	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.624592E+05	4.278061E+02	0.000000E+00
N=15	-3.917008E-07	-1.337888E-01	0.000000E+00	-8.966662E-06	3.415886E+05	3.876130E+02	0.000000E+00
N=16	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.478434E+05	6.523403E+01	0.000000E+00
N=17	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.396868E+05	5.739113E+01	0.000000E+00
N=18	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.377535E+05	6.132455E+01	0.000000E+00
N=19	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	3.437572E+05	6.130070E+01	0.000000E+00
N=20	4.263467E+00	1.362263E+06	-2.719905E-01	2.484000E+00	3.195200E+05	3.800000E+01	3.800000E+01
N=21	-4.263469E+00	-1.550381E+06	0.000000E+00	-2.826329E+00	3.631208E+05	7.992393E+01	8.942610E+01
N=22	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.463328E+05	3.680315E+02	0.000000E+00
N=23	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.463328E+05	3.680315E+02	0.000000E+00
N=24	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.463328E+05	3.680315E+02	0.000000E+00
N=25	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.463328E+05	3.680315E+02	0.000000E+00
N=26	5.683137E+02	4.178367E+08	-8.148309E-02	1.320000E+00	7.352219E+05	2.820000E+02	2.820000E+02
N=27	-5.683137E+02	-5.238843E+08	0.000000E+00	-1.373542E+00	9.218225E+05	4.268527E+02	4.268527E+02
N=28	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.528815E+05	3.727869E+02	0.000000E+00
N=29	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.829219E+05	4.750696E+02	0.000000E+00
N=30	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.891345E+05	4.912666E+02	0.000000E+00
N=31	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.958087E+05	4.064860E+02	0.000000E+00
N=32	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	9.077018E+05	4.160625E+02	0.000000E+00
N=33	0.000000E+00	0.000000E+00	0.000000E+00	0.000000E+00	8.372189E+05	3.602037E+02	0.000000E+00

***** MASS BALANCE *****

MASS ACCUMULATION RATE:

NET MASS CONVECTED INTO CONFIGURATION:

DMASDT= 0.000000E+00 (KG/SEC)
FLOSUM=-3.861405E-06 (KG/SEC)

GLOBAL CONSERVATION OF MASS:

(DMASDT-FLOSUM) = 3.861405E-06 (KG/SEC) DLSUM= 4.079309E-06

***** FLUID ENERGY BALANCE *****

ENERGY ACCUMULATION RATE:

NET ENERGY CONVECTED INTO CONFIGURATION:

NET ENERGY CONDUCTED INTO CONFIGURATION:

ENERGY TRANSFERRED TO/GENERATED WITHIN FLUID:

ENERGY EXCESS DUE TO MASS RESIDUE:

DEDT= 0.000000E+00 (WATT)
EINSUM=-1.062358E-08 (WATT)
QINSUM=-3.534736E-01 (WATT)
QSOURC= 1.062361E+08 (WATT)
EDLSUM= 2.251271E+00 (WATT)

GLOBAL CONSERVATION OF ENERGY: (IDET-EINSUM-QINSUM-QSOURC-EDLSUM)=-3.566887E+02 (WATT),

***** FLUID INVENTORY *****

FLUID	MASS (KG)	ENERGY (J)	HL (J/KG)	TL (C)
1	4.672897E+04	4.2891567776E+10	9.178796E+05	4.237552E+02
2	4.865227E+00	1.5089430791E+06	3.101486E+05	3.272088E+02
3	2.020761E+01	6.9340465501E+06	3.431403E+05	6.071182E+01
TOTAL	4.2900010766E+10			

***** THERMAL STRUCTURE HEAT BALANCE *****

STRUCTURE NUMBER	HEAT GENERATION	ENERGY TRANSFER AT SURFACE TO FLUID (W)--	NET HEAT SOURCE	STORED ENERGY	TEMPERATURE
	IN SOLID (W)	SIDE=OUT	IN SOLID (W)	.GT.TEMP0(1) (J)	AVG (C)
1	1.062361E+08	1.062361E+08	1.062361E+08	2.1037910089E+08	5.391664E+02
2	0.000000E+00	1.060476E+08	7.152557E-06	9.01233335003E+07	3.906478E+02
3	0.000000E+00	4.820362E+03	-1.836552E+05	8.6650632739E+08	3.900110E+02
4	0.000000E+00	1.129797E+05	-1.081593E+05	-6.3299271669E+08	2.714871E+02
5	0.000000E+00	0.000000E+00	7.549489E+04	-9.19332966614E+08	1.880787E+02
TOTAL	1.062361E+08	0.000000E+00	9.768343E-01	-3.85311361955E+08	3.099330E+02

***** RADIATIVE SURFACES *****

SURFACE NUMBER	TEMPERATURE (CELSIUS)	HEAT TRANSFER TO SURF (WATT)
1	3.918480E+02	-1.836552E+05
2	2.763462E+02	1.836546E+05
3	2.676990E+02	-7.549529E+04
4	1.883771E+02	7.549490E+04

UL (M/S) : VELOCITY IN X-DIRECTION AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

K I-->	1	2	3	4	5	6	7	8
12	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	1.33489E-01	0.00000E+00	8.63589E-03	0.00000E+00	4.77743E-03	0.00000E+00	0.00000E+00	0.00000E+00
10	1.63672E-02	0.00000E+00	1.09303E-03	0.00000E+00	-5.05950E-03	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	3.16071E-04	0.00000E+00	-2.05746E-03	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	1.16955E-03	0.00000E+00	-3.18720E-03	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	6.90609E-04	0.00000E+00	-2.95658E-03	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	5.68241E-04	0.00000E+00	-3.19913E-03	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	7.46763E-05	0.00000E+00	-1.59190E-03	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	1.16075E-03	0.00000E+00	-1.75909E-03	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	5.25181E-03	0.00000E+00	-1.62927E-03	0.00000E+00	0.00000E+00	0.00000E+00
2	-8.53122E-01	0.00000E+00	1.46489E-02	0.00000E+00	-2.78170E-04	0.00000E+00	0.00000E+00	0.00000E+00
1	-1.33308E+00	0.00000E+00	-1.68960E-02	0.00000E+00	6.02527E-04	0.00000E+00	0.00000E+00	0.00000E+00

UL (M/S) : VELOCITY IN X-DIRECTION AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

```

K I--> 1 2 3 4 5 6 7 8
12 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
11 1.4207E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
10 2.9875E-02 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
9 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
8 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
7 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
6 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
5 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
4 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
3 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
2 -9.1500E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
1 -9.6676E-01 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

```

UL (M/S) : VELOCITY IN X-DIRECTION AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 3 *****

```

K I--> 1 2 3 4 5 6 7 8
12 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
11 2.5496E-01 0.0000E+00 8.4237E-03 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
10 1.4719E-01 0.0000E+00 3.0492E-03 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
9 0.0000E+00 0.0000E+00 5.8556E-04 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
8 0.0000E+00 0.0000E+00 3.2301E-04 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
7 0.0000E+00 0.0000E+00 1.8053E-04 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
6 0.0000E+00 0.0000E+00 3.5235E-04 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
5 0.0000E+00 0.0000E+00 1.8536E-04 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
4 0.0000E+00 0.0000E+00 1.2117E-03 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
3 0.0000E+00 0.0000E+00 5.3318E-03 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
2 -9.6922E-01 0.0000E+00 1.4777E-02 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
1 -8.1286E-01 0.0000E+00 1.6948E-02 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

```

VL (M/S) : VELOCITY IN Y-DIRECTION AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

```

K I--> 1 2 3 4 5 6 7 8
12 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
11 2.4037E-02 1.1452E-02 5.2889E-02 5.2216E-02 6.2575E-05 5.9151E-05 0.0000E+00 0.0000E+00
10 2.4089E-02 5.8243E-02 3.9929E-02 4.0743E-02 6.5102E-05 6.3623E-05 0.0000E+00 0.0000E+00
9 2.2677E-03 3.0114E-01 2.9584E-03 4.7424E-03 9.4044E-05 9.2750E-05 0.0000E+00 0.0000E+00
8 6.3370E-06 0.0000E+00 2.6457E-02 2.4719E-02 8.3888E-05 8.2496E-05 0.0000E+00 0.0000E+00
7 -1.0644E-04 0.0000E+00 1.8944E-02 1.9125E-02 9.4313E-05 9.3749E-05 0.0000E+00 0.0000E+00
6 0.0000E+00 0.0000E+00 7.5992E-03 8.2663E-03 8.6841E-05 8.6414E-05 0.0000E+00 0.0000E+00
5 0.0000E+00 0.0000E+00 1.6727E-03 1.9978E-03 6.0529E-05 6.0071E-05 0.0000E+00 0.0000E+00
4 0.0000E+00 0.0000E+00 5.8092E-04 3.9961E-04 3.4446E-05 3.4128E-05 0.0000E+00 0.0000E+00
3 -2.8360E-01 0.0000E+00 1.1750E-03 1.0629E-03 1.4459E-05 1.4313E-05 0.0000E+00 0.0000E+00
2 3.2933E-01 9.1666E-01 1.2732E-03 1.2241E-03 5.4789E-06 5.4292E-06 0.0000E+00 0.0000E+00
1 7.4443E-01 1.6332E+00 1.1360E-03 1.1209E-03 2.7321E-06 2.7182E-06 0.0000E+00 0.0000E+00

```

VL (M/S) : VELOCITY IN Y-DIRECTION AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 2 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		4.54298E-02	9.08205E-03	9.08502E-02	8.86742E-02	2.86338E-04	2.75360E-04	0.00000E+00	0.00000E+00
10		4.42586E-02	1.40184E-01	9.01759E-03	9.31712E-02	7.98938E-04	2.97563E-04	0.00000E+00	0.00000E+00
9		3.52294E-03	0.00000E+00	8.71759E-03	1.40235E-02	2.44265E-04	2.42412E-04	0.00000E+00	0.00000E+00
8		-1.78721E-05	0.00000E+00	6.03367E-02	5.73571E-02	1.96073E-04	1.92335E-04	0.00000E+00	0.00000E+00
7		-1.85522E-04	0.00000E+00	3.87592E-02	3.95609E-02	2.04011E-04	2.01491E-04	0.00000E+00	0.00000E+00
6		0.00000E+00	0.00000E+00	1.04693E-02	1.20789E-02	1.77025E-04	1.74757E-04	0.00000E+00	0.00000E+00
5		0.00000E+00	0.00000E+00	9.09181E-04	1.50050E-03	1.08839E-04	1.07583E-04	0.00000E+00	0.00000E+00
4		0.00000E+00	0.00000E+00	2.55063E-03	2.18314E-03	6.05603E-05	5.96938E-05	0.00000E+00	0.00000E+00
3		-3.01213E-01	1.05121E+00	4.36221E-03	3.99416E-03	2.54610E-05	2.50549E-05	0.00000E+00	0.00000E+00
2		3.42676E-01	3.62481E-01	5.07541E-03	4.82634E-03	9.61004E-06	9.47418E-06	0.00000E+00	0.00000E+00
1		7.86850E-01	1.30634E+00	4.64397E-03	4.56897E-03	4.76190E-06	4.73127E-06	0.00000E+00	0.00000E+00

WL (M/S) : VELOCITY IN Z-DIRECTION AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		0.00000E+00	1.35532E-09	0.00000E+00	8.96598E-06	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10		2.64931E-01	6.20591E-02	1.27776E-01	2.08767E-01	3.15114E+00	2.47968E+00	0.00000E+00	0.00000E+00
9		3.71217E-01	0.00000E+00	5.89712E-02	2.10974E-01	3.07742E+00	2.49136E+00	0.00000E+00	0.00000E+00
8		3.84479E-01	5.42838E+00	4.06172E-02	1.95456E-01	2.99260E+00	2.47686E+00	0.00000E+00	0.00000E+00
7		3.84515E-01	5.42836E+00	1.23369E-01	1.90318E-01	2.89503E+00	2.47526E+00	0.00000E+00	0.00000E+00
6		6.12264E-05	5.42833E+00	1.76685E-01	1.79646E-01	2.80071E+00	2.47108E+00	0.00000E+00	0.00000E+00
5		6.12239E+00	5.42830E+00	2.00247E-01	1.81730E-01	2.72318E+00	2.46988E+00	0.00000E+00	0.00000E+00
4		5.88804E+00	5.42828E+00	1.95021E-01	1.73697E-01	2.64806E+00	2.45078E+00	0.00000E+00	0.00000E+00
3		5.88772E+00	5.42825E+00	1.56698E-01	1.43972E-01	2.59209E+00	2.43830E+00	0.00000E+00	0.00000E+00
2		-1.06086E-01	5.42824E+00	9.72966E-02	9.16798E-02	2.56992E+00	2.43285E+00	0.00000E+00	0.00000E+00
1		-1.08890E-01	3.91940E-01	5.17151E-02	4.90448E-02	2.56477E+00	2.42993E+00	0.00000E+00	0.00000E+00

WL (M/S) : VELOCITY IN Z-DIRECTION AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 2 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10		2.66087E-01	8.33745E-02	1.44985E-01	1.97065E-01	3.15191E+00	2.47971E+00	0.00000E+00	0.00000E+00
9		3.77196E-01	0.00000E+00	6.56146E-02	2.11093E-01	3.07774E+00	2.49141E+00	0.00000E+00	0.00000E+00
8		3.84536E-01	1.65017E+01	4.17610E-02	1.98295E-01	2.99269E+00	2.47686E+00	0.00000E+00	0.00000E+00
7		3.84393E-01	1.65040E+01	1.35078E-01	1.79775E-01	2.89497E+00	2.47527E+00	0.00000E+00	0.00000E+00
6		6.12325E+00	1.65039E+01	1.90134E-01	1.67642E-01	2.80062E+00	2.47108E+00	0.00000E+00	0.00000E+00
5		6.12300E+00	6.20547E-01	2.67734E-01	1.75355E-01	2.72312E+00	2.46987E+00	0.00000E+00	0.00000E+00
4		5.88866E+00	6.20547E-01	1.99644E-01	1.71069E-01	2.64803E+00	2.45078E+00	0.00000E+00	0.00000E+00
3		5.88833E+00	6.20547E-01	1.58595E-01	1.42605E-01	2.59208E+00	2.43830E+00	0.00000E+00	0.00000E+00
2		3.39658E-01	0.00000E+00	9.80505E-02	9.11634E-02	2.56992E+00	2.43285E+00	0.00000E+00	0.00000E+00
1		1.62168E-01	2.16346E-02	5.20666E-02	4.87732E-02	2.56477E+00	2.42993E+00	0.00000E+00	0.00000E+00

WL (M/S) : VELOCITY IN Z-DIRECTION AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 3 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10		2.65607E-01	1.52040E-01	2.14342E-01	1.11146E-01	3.14781E+00	2.47945E+00	0.00000E+00	0.00000E+00
9		3.93859E-01	3.15653E+00	3.06358E-01	1.24935E-01	3.07453E+00	2.49105E+00	0.00000E+00	1.37377E+00
8		3.83556E-01	3.11619E+00	3.15474E-01	1.24574E-01	2.99004E+00	2.47662E+00	0.00000E+00	1.35578E+00
7		3.83607E-01	3.07597E+00	2.43228E-01	1.52372E-01	2.89330E+00	2.47519E+00	0.00000E+00	1.33791E+00
6		6.11673E+00	3.03601E+00	2.04726E-01	1.72817E-01	2.79959E+00	2.47104E+00	0.00000E+00	1.32023E+00
5		6.12648E+00	3.10279E-01	2.05879E-01	1.87253E-01	2.72267E+00	2.46981E+00	0.00000E+00	0.00000E+00
4		5.89215E+00	3.10276E-01	1.93284E-01	1.79973E-01	2.64788E+00	2.45077E+00	0.00000E+00	0.00000E+00
3		5.89182E+00	3.10274E-01	1.55635E-01	1.47504E-01	2.59205E+00	2.43830E+00	0.00000E+00	0.00000E+00
2		6.21795E-01	0.00000E+00	9.68778E-02	9.31061E-02	2.56991E+00	2.43285E+00	0.00000E+00	0.00000E+00
1		3.49756E-01	4.73247E-02	5.14167E-02	4.96335E-02	2.56477E+00	2.42993E+00	0.00000E+00	0.00000E+00

TL (DEGREES CELSIUS) : TEMPERATURE AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	4.99078E+02	0.00000E+00	3.87613E+02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		4.99116E+02	4.99072E+02	4.52718E+02	3.87613E+02	8.56693E+01	7.32018E+01	0.00000E+00	0.00000E+00
10		4.99110E+02	4.99024E+02	4.26892E+02	3.48344E+02	8.25692E+01	7.01195E+01	0.00000E+00	0.00000E+00
9		4.99101E+02	3.60265E+02	3.43385E+02	3.13417E+02	7.84964E+01	6.70796E+01	0.00000E+00	0.00000E+00
8		4.99090E+02	3.60250E+02	3.43775E+02	2.99496E+02	7.17515E+01	6.20231E+01	0.00000E+00	0.00000E+00
7		4.99080E+02	3.60231E+02	3.42880E+02	2.90170E+02	6.50417E+01	5.70408E+01	0.00000E+00	0.00000E+00
6		4.99053E+02	3.60214E+02	3.39890E+02	2.83958E+02	5.82669E+01	5.21264E+01	0.00000E+00	0.00000E+00
5		4.98911E+02	3.60196E+02	3.33914E+02	2.76183E+02	5.28024E+01	4.82524E+01	0.00000E+00	0.00000E+00
4		3.60369E+02	3.60181E+02	3.18973E+02	2.70383E+02	4.60854E+01	4.35671E+01	0.00000E+00	0.00000E+00
3		3.60168E+02	3.60173E+02	2.97067E+02	2.66931E+02	4.11656E+01	4.01753E+01	0.00000E+00	0.00000E+00
2		3.60168E+02	3.60167E+02	2.81652E+02	2.65356E+02	3.91688E+01	3.88056E+01	0.00000E+00	0.00000E+00
1		3.60168E+02	3.60165E+02	2.76880E+02	2.63786E+02	3.85825E+01	3.84029E+01	0.00000E+00	0.00000E+00

TL (DEGREES CELSIUS) : TEMPERATURE AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 2 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		4.99107E+02	4.99066E+02	4.55776E+02	3.89336E+02	8.67215E+01	7.32017E+01	0.00000E+00	0.00000E+00
10		4.99098E+02	4.99029E+02	4.31142E+02	3.50386E+02	8.25970E+01	7.01194E+01	0.00000E+00	0.00000E+00
9		4.99087E+02	3.60273E+02	3.46375E+02	3.14857E+02	7.85120E+01	6.70796E+01	0.00000E+00	0.00000E+00
8		4.99075E+02	3.60268E+02	3.44311E+02	3.00102E+02	7.17535E+01	6.20231E+01	0.00000E+00	0.00000E+00
7		4.99065E+02	3.60783E+02	3.43476E+02	2.90470E+02	6.50362E+01	5.70408E+01	0.00000E+00	0.00000E+00
6		4.99038E+02	3.60756E+02	3.40322E+02	2.84093E+02	5.82640E+01	5.21264E+01	0.00000E+00	0.00000E+00
5		4.98896E+02	3.60753E+02	3.34155E+02	2.76198E+02	5.28008E+01	4.82524E+01	0.00000E+00	0.00000E+00
4		3.60369E+02	3.60752E+02	3.19135E+02	2.70373E+02	4.60848E+01	4.35671E+01	0.00000E+00	0.00000E+00
3		3.60168E+02	3.60752E+02	2.97134E+02	2.66911E+02	4.11654E+01	4.01753E+01	0.00000E+00	0.00000E+00
2		3.60168E+02	3.60166E+02	2.81635E+02	2.65331E+02	3.91688E+01	3.88056E+01	0.00000E+00	0.00000E+00
1		3.60167E+02	3.60165E+02	2.76904E+02	2.63755E+02	3.85825E+01	3.84029E+01	0.00000E+00	0.00000E+00

TL (DEGREES CELSIUS) : TEMPERATURE AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 3 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
11		4.9905E+02	4.99029E+02	4.5614E+02	3.93667E+02	8.65950E+01	7.32031E+01	0.0000E+00	0.0000E+00
10		4.99027E+02	4.99014E+02	4.37834E+02	3.60719E+02	8.25217E+01	7.01205E+01	0.0000E+00	4.26853E+02
9		4.99005E+02	4.54080E+02	4.05991E+02	3.23284E+02	7.84426E+01	6.70803E+01	0.0000E+00	4.26826E+02
8		4.98992E+02	4.07981E+02	3.68575E+02	2.99218E+02	7.16661E+01	6.20233E+01	0.0000E+00	3.79595E+02
7		4.98981E+02	3.60832E+02	3.42970E+02	2.86741E+02	6.49785E+01	5.70409E+01	0.0000E+00	3.31275E+02
6		4.98954E+02	3.60812E+02	3.39953E+02	2.81410E+02	5.82380E+01	5.21265E+01	0.0000E+00	2.82002E+02
5		4.98812E+02	3.60766E+02	3.33483E+02	2.74999E+02	5.27903E+01	4.82524E+01	0.0000E+00	0.0000E+00
4		3.60367E+02	3.60655E+02	3.18699E+02	2.69771E+02	4.60817E+01	4.35671E+01	0.0000E+00	0.0000E+00
3		3.60166E+02	3.60754E+02	2.96731E+02	2.66507E+02	4.11646E+01	4.01753E+01	0.0000E+00	0.0000E+00
2		3.60164E+02	3.60157E+02	2.81260E+02	2.64991E+02	3.91685E+01	3.88056E+01	0.0000E+00	0.0000E+00
1		3.60164E+02	3.60160E+02	2.76563E+02	2.63477E+02	3.85824E+01	3.84029E+01	0.0000E+00	0.0000E+00

PRESSURE (PASCAL) AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.0000E+00	1.01300E+05	0.0000E+00	1.01300E+05	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
11		1.09718E+05	1.09703E+05	1.01307E+05	1.01307E+05	1.01300E+05	1.01300E+05	0.0000E+00	0.0000E+00
10		1.18337E+05	1.18338E+05	1.01315E+05	1.01315E+05	1.01311E+05	1.01311E+05	0.0000E+00	0.0000E+00
9		1.29918E+05	7.62850E+05	1.01326E+05	1.01326E+05	1.01326E+05	1.01326E+05	0.0000E+00	0.0000E+00
8		1.44439E+05	7.71971E+05	1.01340E+05	1.01340E+05	1.01345E+05	1.01345E+05	0.0000E+00	0.0000E+00
7		1.58961E+05	7.86524E+05	1.01354E+05	1.01354E+05	1.01364E+05	1.01364E+05	0.0000E+00	0.0000E+00
6		1.95812E+05	7.99606E+05	1.01367E+05	1.01367E+05	1.01381E+05	1.01381E+05	0.0000E+00	0.0000E+00
5		3.86868E+05	8.12484E+05	1.01380E+05	1.01380E+05	1.01398E+05	1.01398E+05	0.0000E+00	0.0000E+00
4		5.62234E+05	8.24707E+05	1.01392E+05	1.01392E+05	1.01415E+05	1.01415E+05	0.0000E+00	0.0000E+00
3		8.33004E+05	8.33194E+05	1.01400E+05	1.01400E+05	1.01425E+05	1.01425E+05	0.0000E+00	0.0000E+00
2		8.35704E+05	8.39277E+05	1.01403E+05	1.01403E+05	1.01429E+05	1.01429E+05	0.0000E+00	0.0000E+00
1		8.37028E+05	8.40649E+05	1.01404E+05	1.01404E+05	1.01431E+05	1.01431E+05	0.0000E+00	0.0000E+00

PRESSURE (PASCAL) AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 2 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
11		1.09717E+05	1.09703E+05	1.01307E+05	1.01307E+05	1.01300E+05	1.01300E+05	0.0000E+00	0.0000E+00
10		1.18335E+05	1.18333E+05	1.01315E+05	1.01315E+05	1.01311E+05	1.01311E+05	0.0000E+00	0.0000E+00
9		1.29918E+05	7.62728E+05	1.01326E+05	1.01326E+05	1.01326E+05	1.01326E+05	0.0000E+00	0.0000E+00
8		1.44439E+05	7.80770E+05	1.01340E+05	1.01340E+05	1.01345E+05	1.01345E+05	0.0000E+00	0.0000E+00
7		1.58961E+05	7.74756E+05	1.01354E+05	1.01354E+05	1.01364E+05	1.01364E+05	0.0000E+00	0.0000E+00
6		1.95818E+05	7.74713E+05	1.01367E+05	1.01367E+05	1.01381E+05	1.01381E+05	0.0000E+00	0.0000E+00
5		3.86907E+05	7.85875E+05	1.01380E+05	1.01380E+05	1.01398E+05	1.01398E+05	0.0000E+00	0.0000E+00
4		5.62304E+05	7.70364E+05	1.01392E+05	1.01392E+05	1.01415E+05	1.01415E+05	0.0000E+00	0.0000E+00
3		8.33003E+05	7.79795E+05	1.01400E+05	1.01400E+05	1.01425E+05	1.01425E+05	0.0000E+00	0.0000E+00
2		8.35996E+05	8.35742E+05	1.01403E+05	1.01403E+05	1.01429E+05	1.01429E+05	0.0000E+00	0.0000E+00

1 8.37265E+05 6.39885E+05 1.01404E+05 1.01404E+05 1.01431E+05 1.01431E+05 0.00000E+00 0.00000E+00

PRESSURE (PASCAL) AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 3 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		1.09715E+05	1.09708E+05	1.01308E+05	1.01308E+05	1.01300E+05	1.01300E+05	0.00000E+00	0.00000E+00
10		1.08329E+05	1.18303E+05	1.01315E+05	1.01315E+05	1.01311E+05	1.01311E+05	0.00000E+00	1.01300E+05
9		1.29917E+05	1.22603E+05	1.01326E+05	1.01326E+05	1.01326E+05	1.01326E+05	0.00000E+00	1.37193E+05
8		1.44439E+05	1.30141E+05	1.01340E+05	1.01340E+05	1.01345E+05	1.01345E+05	0.00000E+00	1.81690E+05
7		1.58961E+05	1.37853E+05	1.01354E+05	1.01354E+05	1.01364E+05	1.01364E+05	0.00000E+00	2.26021E+05
6		1.95845E+05	1.46204E+05	1.01367E+05	1.01367E+05	1.01381E+05	1.01381E+05	0.00000E+00	2.65674E+05
5		3.87108E+05	1.59563E+05	1.01380E+05	1.01380E+05	1.01398E+05	1.01398E+05	0.00000E+00	0.00000E+00
4		5.62688E+05	1.72244E+05	1.01392E+05	1.01392E+05	1.01415E+05	1.01415E+05	0.00000E+00	0.00000E+00
3		8.33356E+05	1.79751E+05	1.01400E+05	1.01400E+05	1.01425E+05	1.01425E+05	0.00000E+00	0.00000E+00
2		8.36331E+05	8.38142E+05	1.01403E+05	1.01403E+05	1.01429E+05	1.01429E+05	0.00000E+00	0.00000E+00
1		8.37723E+05	8.39420E+05	1.01404E+05	1.01404E+05	1.01431E+05	1.01431E+05	0.00000E+00	0.00000E+00

RL (KG/M**3) : DENSITY AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	8.32471E+02	0.00000E+00	7.35733E-01	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		8.32462E+02	8.32472E+02	6.69793E-01	7.35787E-01	9.75204E-01	1.01312E+00	0.00000E+00	0.00000E+00
10		8.32463E+02	8.32484E+02	6.94553E-01	7.82333E-01	9.86551E-01	1.02233E+00	0.00000E+00	0.00000E+00
9		8.32465E+02	8.65670E+02	7.88710E-01	8.29005E-01	9.98123E-01	1.03162E+00	0.00000E+00	0.00000E+00
8		8.32468E+02	8.65673E+02	7.88321E-01	8.49275E-01	1.01783E+00	1.04737E+00	0.00000E+00	0.00000E+00
7		8.32470E+02	8.65678E+02	7.89578E-01	8.63457E-01	1.03822E+00	1.06338E+00	0.00000E+00	0.00000E+00
6		8.32477E+02	8.65682E+02	7.93531E-01	8.73197E-01	1.05962E+00	1.07963E+00	0.00000E+00	0.00000E+00
5		8.32511E+02	8.65686E+02	8.01444E-01	8.85668E-01	1.07757E+00	1.09283E+00	0.00000E+00	0.00000E+00
4		8.65645E+02	8.65689E+02	8.21768E-01	8.95229E-01	1.10043E+00	1.11091E+00	0.00000E+00	0.00000E+00
3		8.65693E+02	8.65691E+02	8.53400E-01	9.01018E-01	1.11766E+00	1.12129E+00	0.00000E+00	0.00000E+00
2		8.65693E+02	8.65693E+02	8.77137E-01	9.03678E-01	1.12495E+00	1.12626E+00	0.00000E+00	0.00000E+00
1		8.65693E+02	8.65693E+02	8.84757E-01	9.06334E-01	1.12708E+00	1.12773E+00	0.00000E+00	0.00000E+00

PL (KG/M**3) : DENSITY AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 2 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		8.32464E+02	8.32474E+02	6.69893E-01	7.33873E-01	9.75062E-01	1.01312E+00	0.00000E+00	0.00000E+00
10		8.32466E+02	8.32483E+02	6.90363E-01	7.79772E-01	9.86473E-01	1.02233E+00	0.00000E+00	0.00000E+00
9		8.32469E+02	8.65668E+02	7.84904E-01	8.26975E-01	9.98079E-01	1.03162E+00	0.00000E+00	0.00000E+00
8		8.32472E+02	8.65669E+02	7.87636E-01	8.48377E-01	1.01782E+00	1.04737E+00	0.00000E+00	0.00000E+00
7		8.32474E+02	8.65546E+02	7.88814E-01	8.62997E-01	1.03823E+00	1.06338E+00	0.00000E+00	0.00000E+00
6		8.32481E+02	8.65553E+02	7.92297E-01	8.72985E-01	1.05963E+00	1.07963E+00	0.00000E+00	0.00000E+00
5		8.32514E+02	8.65553E+02	8.01126E-01	8.85644E-01	1.07758E+00	1.09283E+00	0.00000E+00	0.00000E+00
4		8.65645E+02	8.65554E+02	8.21543E-01	8.95245E-01	1.10043E+00	1.11091E+00	0.00000E+00	0.00000E+00

```

3      8.65693E+02 8.65554E+02 8.53300E-01 9.01052E-01 1.11765E+00 1.12129E+00 0.00000E+00 0.00000E+00
4      8.65693E+02 8.65693E+02 8.77162E-01 9.03720E-01 1.12495E+00 1.12626E+00 0.00000E+00 0.00000E+00
5      8.65693E+02 8.65693E+02 8.84719E-01 9.06384E-01 1.12708E+00 1.12773E+00 0.00000E+00 0.00000E+00

```

RL (KG/M**3) : DENSITY AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 3 *****

```

K I--> 1 2 3 4 5 6 7 8
12      0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
11      8.32478E+02 8.32483E+02 6.66648E-01 7.29108E-01 9.75405E-01 1.01312E+00 0.00000E+00 0.00000E+00
10      8.32483E+02 8.32486E+02 6.83866E-01 7.67063E-01 9.86682E-01 1.02233E+00 0.00000E+00 8.49780E+02
9      8.32488E+02 8.43263E+02 7.16005E-01 8.15291E-01 9.98276E-01 1.03161E+00 0.00000E+00 8.49786E+02
8      8.32492E+02 8.54290E+02 7.57855E-01 8.49686E-01 1.01808E+00 1.04737E+00 0.00000E+00 8.61064E+02
7      8.32494E+02 8.65535E+02 7.89461E-01 8.68744E-01 1.03841E+00 1.06338E+00 0.00000E+00 8.72564E+02
6      8.32501E+02 8.65539E+02 7.93448E-01 8.77208E-01 1.05972E+00 1.07963E+00 0.00000E+00 8.84248E+02
5      8.32535E+02 8.65545E+02 8.02014E-01 8.87581E-01 1.07761E+00 1.09283E+00 0.00000E+00 0.00000E+00
4      8.65645E+02 8.65551E+02 8.22147E-01 8.96238E-01 1.10044E+00 1.10917E+00 0.00000E+00 0.00000E+00
3      8.65693E+02 8.65553E+02 8.53904E-01 9.01727E-01 1.11776E+00 1.12129E+00 0.00000E+00 0.00000E+00
2      8.65694E+02 8.65695E+02 8.7756E-01 9.04292E-01 1.12495E+00 1.12626E+00 0.00000E+00 0.00000E+00
1      8.65694E+02 8.65694E+02 8.85267E-01 9.06855E-01 1.12708E+00 1.12773E+00 0.00000E+00 0.00000E+00

```

DL (KG/M**3/SEC) : MASS RESIDUAL AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

```

K I--> 1 2 3 4 5 6 7 8
12      0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
11      5.34992E-08 2.65599E-07 1.73716E-08 6.59095E-08 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
10      1.26465E-06 3.34404E-07 8.16639E-08 3.35122E-08 1.64908E-07 2.19380E-08 0.00000E+00 0.00000E+00
9      -1.20763E-07 5.92839E-08 5.25578E-08 2.93821E-09 4.10178E-09 1.01987E-07 0.00000E+00 0.00000E+00
8      2.73392E-07 2.58620E-07 2.26046E-08 3.95083E-08 8.03279E-08 6.96633E-09 0.00000E+00 0.00000E+00
7      -1.12242E-09 2.88753E-07 1.64234E-08 3.94533E-08 5.69554E-08 1.65811E-07 0.00000E+00 0.00000E+00
6      -2.03293E-10 8.04504E-07 1.22441E-08 5.72554E-08 1.89941E-08 1.97716E-07 0.00000E+00 0.00000E+00
5      1.07274E-07 2.65840E-08 2.65738E-08 4.38545E-08 5.21785E-08 1.04726E-07 0.00000E+00 0.00000E+00
4      -2.59363E-07 1.26991E-06 6.55099E-08 5.13445E-08 1.65748E-07 8.22418E-08 0.00000E+00 0.00000E+00
3      9.83420E-07 4.58045E-06 3.41821E-07 1.30427E-07 5.98514E-07 2.94797E-07 0.00000E+00 0.00000E+00
2      1.70868E-06 4.69969E-06 2.87797E-06 9.66618E-07 2.26132E-06 3.88775E-06 0.00000E+00 0.00000E+00
1      5.29320E-07 1.45840E-06 1.32666E-07 2.76762E-06 1.26507E-07 2.55921E-06 0.00000E+00 0.00000E+00

```

DL (KG/M**3/SEC) : MASS RESIDUAL AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 2 *****

```

K I--> 1 2 3 4 5 6 7 8
12      0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
11      3.40694E-06 1.25791E-06 2.06443E-08 5.75059E-08 0.00000E+00 0.0000E+00 0.0000E+00 0.0000E+00
10      -5.35465E-06 2.26857E-06 7.81338E-08 9.51574E-08 1.20536E-07 5.11587E-08 0.00000E+00 0.0000E+00
9      -5.85675E-08 1.34225E-07 4.80024E-08 4.12385E-08 1.09168E-07 1.36834E-08 0.00000E+00 0.0000E+00
8      1.34492E-07 3.90128E-06 5.2628E-08 3.22326E-08 1.6001E-07 2.47170E-08 0.00000E+00 0.0000E+00
7      8.4497E-06 5.9324E-07 5.6256E-08 1.25107E-08 2.0610E-07 7.56040E-08 0.00000E+00 0.0000E+00
6      -3.91801E-06 1.70921E-07 4.43533E-06 4.42106E-08 1.24090E-07 1.80707E-07 0.0000E+00 0.0000E+00

```

5 1.10335E-07-5.83101E-08 5.13521E-06 4.08166E-03 1.03731E-07 1.14806E-07 5.00000E+00 0.00000E+00
 4 -3.29353E-07-1.86202E-07 1.00000E-07 4.21851E-08 2.86423E-07 3.24068E-09 0.00000E+00 0.00000E+00
 3 -1.88526E-06 3.59656E-08 3.81053E-07 1.12106E-07 8.18676E-07 1.33649E-07 0.00000E+00 0.00000E+00
 2 4.98589E-06-1.06349E-06 2.84582E-06 9.22700E-07 2.42495E-06 3.94860E-06 0.00000E+00 0.00000E+00
 1 -1.62877E-06 1.51460E-07 1.33133E-07 2.70623E-06 1.06250E-07 2.72355E-06 0.00000E+00 0.00000E+00

DL (KG/M**3/SEC) : MASS RESIDUAL AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 3 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		-1.61863E-06	8.77373E-07	3.77602E-08	3.00324E-09	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10		-2.64974E-06	3.20719E-07	1.97760E-08	3.54846E-08	8.02467E-09	1.04632E-08	0.00000E+00	0.00000E+00
9		-2.56805E-07	2.78743E-07	3.86474E-09	1.53959E-08	-8.12386E-09	4.93767E-09	0.00000E+00	2.41639E-09
8		-1.75395E-07	9.26828E-08	4.22460E-09	5.93881E-09	4.27054E-08	-8.26823E-10	0.00000E+00	5.15784E-10
7		4.96057E-09	-2.88948E-08	2.12022E-08	-1.42979E-11	8.99227E-08	1.29329E-08	0.00000E+00	2.25066E-09
6		-7.17893E-08	3.11907E-09	2.82066E-08	8.04615E-09	9.91316E-08	5.08405E-08	0.00000E+00	2.65308E-09
5		1.10796E-07	-1.67714E-08	2.90319E-08	1.19276E-08	8.59299E-08	4.98370E-08	0.00000E+00	0.00000E+00
4		-4.08992E-07	2.88801E-07	6.75250E-08	1.70094E-08	2.10900E-07	3.05779E-08	0.00000E+00	0.00000E+00
3		9.12505E-07	-6.32027E-07	3.46315E-07	7.94406E-08	6.42763E-07	1.40657E-07	0.00000E+00	0.00000E+00
2		-1.53676E-06	7.08542E-06	2.79424E-06	8.84977E-07	3.34904E-06	2.37699E-06	0.00000E+00	0.00000E+00
1		-9.50085E-07	7.88283E-06	1.36405E-07	2.64727E-06	-2.26628E-07	3.07396E-05	0.00000E+00	0.00000E+00

QSQUR (WATT/M**3) : VOLUMETRIC HEAT SOURCE AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 1 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		0.00000E+00	-3.30900E+03	1.10737E+03	-2.93832E+03	1.15391E+04	6.80242E+03	0.00000E+00	0.00000E+00
10		0.00000E+00	-3.38613E+03	2.11239E+03	-2.44127E+03	1.13270E+04	6.80242E+03	0.00000E+00	0.00000E+00
9		0.00000E+00	-3.23993E+03	2.07432E+02	-1.28656E+03	1.08343E+04	6.80242E+03	0.00000E+00	0.00000E+00
8		0.00000E+00	-3.24311E+03	2.48851E+02	-9.43887E+02	1.06881E+04	6.80242E+03	0.00000E+00	0.00000E+00
7		0.00000E+00	-3.25134E+03	3.56103E+02	-7.56609E+02	1.06082E+04	6.80242E+03	0.00000E+00	0.00000E+00
6		0.00000E+00	-3.26427E+03	5.24510E+02	-6.88561E+02	1.05792E+04	6.80242E+03	0.00000E+00	0.00000E+00
5		3.25441E+07	-3.28578E+03	8.04876E+02	-5.36579E+02	1.05143E+04	6.80242E+03	0.00000E+00	0.00000E+00
4		0.00000E+00	-3.33197E+03	1.40672E+03	-4.85948E+02	1.04927E+04	6.80242E+03	0.00000E+00	0.00000E+00
3		0.00000E+00	-3.38050E+03	2.03911E+03	-4.50084E+02	1.04774E+04	6.80242E+03	0.00000E+00	0.00000E+00
2		0.00000E+00	-3.38374E+03	2.08121E+03	-3.79742E+02	1.04474E+04	6.80242E+03	0.00000E+00	0.00000E+00
1		0.00000E+00	-3.33529E+03	1.44990E+03	-2.25508E+02	1.03816E+04	6.80242E+03	0.00000E+00	0.00000E+00

QSQUR (WATT/M**3) : VOLUMETRIC HEAT SOURCE AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 2 *****

K	I-->	1	2	3	4	5	6	7	8
12		0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11		0.00000E+00	-3.31611E+03	1.20007E+03	-3.10602E+03	1.16106E+04	6.80242E+03	0.00000E+00	0.00000E+00
10		0.00000E+00	-3.39400E+03	2.21501E+03	-2.52619E+03	1.13632E+04	6.80242E+03	0.00000E+00	0.00000E+00
9		0.00000E+00	-3.23485E+03	1.41190E+02	-1.33868E+03	1.08566E+04	6.80242E+03	0.00000E+00	0.00000E+00
8		0.00000E+00	-3.24337E+03	2.52211E+02	-9.64273E+02	1.06968E+04	6.80242E+03	0.00000E+00	0.00000E+00

```

7 6.00000E+00-3.25253E+03 3.71603E+02-7.54235E+02 1.06072E+04 6.80242E+03 0.00000E+00 0.00000E+00
8 6.00000E+00-3.26573E+03 5.43508E+02-6.81781E+02 1.05763E+04 6.80242E+03 0.00000E+00 0.00000E+00
9 3.25441E+07-3.28679E+03 8.17935E+02-5.32670E+02 1.05127E+04 6.80242E+03 0.00000E+00 0.00000E+00
4 6.00000E+00-3.33324E+03 1.42326E+03-4.83762E+02 1.04918E+04 6.80242E+03 0.00000E+00 0.00000E+00
3 6.00000E+00-3.38197E+03 2.05818E+03-4.48490E+02 1.04767E+04 6.80242E+03 0.00000E+00 0.00000E+00
2 6.00000E+00-3.38394E+03 2.08394E+03-3.78552E+02 1.04469E+04 6.80242E+03 0.00000E+00 0.00000E+00
1 6.00000E+00-3.33585E+03 1.45728E+03-2.25081E+02 1.03814E+04 6.80242E+03 0.00000E+00 0.00000E+00

```

QSOUR (WATT/M**3) : VOLUMETRIC HEAT SOURCE AT TIME= 4.067E+06 SECONDS

***** CONSTANT J PLANE J= 3 *****

```

K I-->      1      2      3      4      5      6      7      8
12 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
11 0.00000E+00-3.32042E+03 1.25618E+03-2.73165E+03 1.14509E+04 6.80242E+03 0.00000E+00 0.00000E+00
10 0.00000E+00-3.43468E+03 2.74509E+03-2.44961E+03 1.13306E+04 6.80242E+03 0.00000E+00 0.00000E+00
9 0.00000E+00-8.57290E+06 2.21132E+03-1.38335E+03 1.08756E+04 6.80242E+03 0.00000E+00 8.42226E+06
8 0.00000E+00-8.84573E+06 1.64233E+03-8.44876E+02 1.06459E+04 6.80242E+03 0.00000E+00 8.69045E+06
7 0.00000E+00-9.11446E+06 4.46151E+02-6.19707E+02 1.05498E+04 6.80242E+03 0.00000E+00 8.95465E+03
6 0.00000E+00-3.26729E+03 5.63866E+02-6.05499E+02 1.05437E+04 6.80242E+03 0.00000E+00 0.00000E+00
5 3.25441E+07-3.28843E+03 8.39383E+02-5.03797E+02 1.05003E+04 6.80242E+03 0.00000E+00 0.00000E+00
4 0.00000E+00-3.33329E+03 1.42394E+03-4.71638E+02 1.04866E+04 6.80242E+03 0.00000E+00 0.00000E+00
3 0.00000E+00-3.38197E+03 2.05825E+03-4.41262E+02 1.04737E+04 6.80242E+03 0.00000E+00 0.00000E+00
2 0.00000E+00-3.38380E+03 2.08209E+03-3.73014E+02 1.04445E+04 6.80242E+03 0.00000E+00 0.00000E+00
1 0.00000E+00-3.33561E+03 1.45415E+03-2.21543E+02 1.03799E+04 6.80242E+03 0.00000E+00 0.00000E+00

```

THERMAL STRUCTURE NUMBER 1 AT TIME= 4.067E+06 SECONDS

```

      FLUID HEAT TRANSFER
ELEMENT CELL TEMPERATURE COEFFICIENT <----- THERMAL STRUCTURE TEMPERATURES ----->
      1 10105 4.9891E+02 2.0468E+05 5.0859E+02 5.3380E+02 5.5518E+02 5.7081E+02 5.8105E+02 5.8612E+02
      2 10205 4.9890E+02 2.0468E+05 5.0857E+02 5.3379E+02 5.5516E+02 5.7080E+02 5.8104E+02 5.8610E+02
      3 10305 4.9881E+02 2.0472E+05 5.0849E+02 5.3370E+02 5.5508E+02 5.7072E+02 5.8096E+02 5.8603E+02

```

THERMAL STRUCTURE NUMBER 2 AT TIME= 4.067E+06 SECONDS

```

      1ST FLUID HEAT TRANSFER 2ND FLUID HEAT TRANSFER
ELEMENT CELL TEMPERATURE COEFFICIENT <----- THERMAL STRUCTURE TEMPERATURES ----->
      4 80307 3.3128E+02 1.2500E+05 20307 3.6083E+02 4.3893E+04 3.3689E+02 3.4283E+02 3.4897E+02
      5 80308 3.7960E+02 1.2247E+05 20308 4.0798E+02 4.2926E+04 3.8501E+02 3.9060E+02 3.9637E+02
      6 80309 4.2683E+02 1.2015E+05 20309 4.5408E+02 4.2024E+04 4.3205E+02 4.3730E+02 4.4274E+02

```

THERMAL STRUCTURE NUMBER 3 AT TIME= 4.067E+06 SECONDS

```

      1ST FLUID HEAT TRANSFER 2ND FLUID HEAT TRANSFER
ELEMENT CELL TEMPERATURE COEFFICIENT <----- THERMAL STRUCTURE TEMPERATURES ----->
      10 30101 2.7688E+02 1.4920E+06 20101 3.6017E+02 1.2015E+04 3.5120E+02 3.5336E+02 3.5770E+02 3.5988E+02
      11 30201 2.7690E+02 1.4991E+06 20201 3.6016E+02 1.4466E+04 3.5125E+02 3.5340E+02 3.5771E+02 3.5993E+02
      12 30301 2.7656E+02 1.4944E+06 20301 3.6016E+02 1.9340E+04 3.5115E+02 3.5331E+02 3.5770E+02 3.5993E+02
      13 30102 2.8465E+02 2.0267E+06 20102 3.6017E+02 2.1097E+04 3.5120E+02 3.5336E+02 3.5779E+02 3.6000E+02

```

14	3202	2.8164E+02	2.2967E+00	20202	3.6017E+02	1.0436E+04	3.5103E+02	3.5321E+02	3.5762E+02	3.5954E+02
15	3202	2.8126E+02	2.2846E+00	20302	3.6016E+02	8.8426E+03	3.5096E+02	3.5315E+02	3.5755E+02	3.5947E+02
16	3203	2.9707E+02	2.8774E+00	20103	3.6017E+02	3.1138E+04	3.5126E+02	3.5345E+02	3.5785E+02	3.6006E+02
17	3203	2.9713E+02	2.8907E+00	20203	3.6075E+02	9.4030E+03	3.5159E+02	3.5377E+02	3.5817E+02	3.6039E+02
18	3203	2.9673E+02	2.8702E+00	20303	3.6075E+02	9.0249E+02	3.5157E+02	3.5376E+02	3.5816E+02	3.6037E+02
19	30104	3.1897E+02	3.3176E+00	20104	3.6018E+02	3.1138E+04	3.5140E+02	3.5355E+02	3.5789E+02	3.6007E+02
20	30204	3.1914E+02	3.3403E+00	20204	3.6075E+02	9.4641E+03	3.5172E+02	3.5387E+02	3.5821E+02	3.6039E+02
21	30304	3.1870E+02	3.3060E+00	20304	3.6076E+02	7.4890E+03	3.5164E+02	3.5379E+02	3.5813E+02	3.6031E+02
22	30105	3.3391E+02	3.4927E+00	20105	3.6020E+02	3.1138E+04	3.5154E+02	3.5366E+02	3.5793E+02	3.6009E+02
23	30205	3.3415E+02	3.5353E+00	20205	3.6075E+02	9.4641E+03	3.5185E+02	3.5397E+02	3.5825E+02	3.6040E+02
24	30305	3.3348E+02	3.5076E+00	20305	3.6079E+02	7.4889E+03	3.5178E+02	3.5391E+02	3.5818E+02	3.6034E+02
25	30106	3.3989E+02	3.4216E+00	20106	3.6021E+02	3.1138E+04	3.5161E+02	3.5372E+02	3.5797E+02	3.6011E+02
26	30206	3.4032E+02	3.5048E+00	20206	3.6076E+02	4.2701E+04	3.5218E+02	3.5429E+02	3.5854E+02	3.6068E+02
27	30306	3.3995E+02	3.5523E+00	20306	3.6081E+02	1.5083E+04	3.5209E+02	3.5420E+02	3.5845E+02	3.6059E+02
28	30107	3.4288E+02	3.1004E+00	20107	3.6023E+02	3.1138E+04	3.5166E+02	3.5376E+02	3.5799E+02	3.6012E+02
29	30207	3.4348E+02	3.2300E+00	20207	3.6078E+02	6.8852E+04	3.5227E+02	3.5437E+02	3.5860E+02	3.6073E+02
30	30307	3.4297E+02	3.6960E+00	20307	3.6083E+02	2.1435E+04	3.5220E+02	3.5431E+02	3.5854E+02	3.6068E+02
31	30108	3.4377E+02	2.4000E+00	20108	3.6025E+02	3.1138E+04	3.5170E+02	3.5380E+02	3.5802E+02	3.6014E+02
32	30208	3.4431E+02	2.5823E+00	20208	3.6027E+02	6.8867E+04	3.5178E+02	3.5388E+02	3.5809E+02	3.6022E+02
33	30308	3.6857E+02	4.0789E+00	20308	4.0798E+02	2.1068E+04	3.5937E+02	4.0147E+02	4.0569E+02	4.0782E+02
34	30109	3.4338E+02	1.9149E+00	20109	3.6027E+02	1.9956E+04	3.5167E+02	3.5376E+02	3.5798E+02	3.6010E+02
35	30209	3.4637E+02	1.9987E+00	20209	3.6027E+02	4.1611E+04	3.5178E+02	3.5387E+02	3.5807E+02	3.6019E+02
36	30309	4.0599E+02	4.2697E+00	20309	4.5408E+02	2.0734E+04	4.4560E+02	4.4766E+02	4.5182E+02	4.5391E+02
37	30110	4.2689E+02	2.5498E+00	20110	4.9902E+02	4.9050E+03	4.9025E+02	4.9225E+02	4.9629E+02	4.9832E+02
38	30210	4.3114E+02	2.8631E+00	20210	4.9903E+02	5.4661E+03	4.9031E+02	4.9232E+02	4.9636E+02	4.9840E+02
39	30310	4.3783E+02	3.9804E+00	20310	4.9901E+02	1.3921E+04	4.9058E+02	4.9261E+02	4.9670E+02	4.9876E+02
40	30111	4.5272E+02	2.2394E+00	20111	4.9907E+02	5.1755E+03	4.9054E+02	4.9249E+02	4.9644E+02	4.9842E+02
41	30211	4.5578E+02	2.6415E+00	20211	4.9907E+02	5.2500E+03	4.9052E+02	4.9248E+02	4.9643E+02	4.9842E+02
42	30311	4.5614E+02	2.7938E+00	20311	4.9903E+02	5.7721E+03	4.9053E+02	4.9249E+02	4.9645E+02	4.9844E+02

THERMAL STRUCTURE NUMBER 4 AT TIME= 4.067E+06 SECONDS

1ST		FLUID	HEAT TRANSFER	2ND	FLUID	HEAT TRANSFER	THERMAL STRUCTURE TEMPERATURES	
ELEMENT		CELL	TEMPERATURE COEFFICIENT	CELL		TEMPERATURE COEFFICIENT	----->	
76	50101	3.8582E+01	8.8480E+00	40101	2.6379E+02	1.4611E+00	2.4347E+02	2.4561E+02
77	50201	3.8582E+01	8.8480E+00	40201	2.6376E+02	1.4616E+00	2.4347E+02	2.4561E+02
78	50301	3.8582E+01	8.8480E+00	40301	2.6348E+02	1.4699E+00	2.4344E+02	2.4558E+02
79	50102	3.9169E+01	8.9146E+00	40102	2.6536E+02	2.2321E+00	2.4381E+02	2.4596E+02
80	50202	3.9169E+01	8.9146E+00	40202	2.6533E+02	2.2277E+00	2.4380E+02	2.4595E+02
81	50302	3.9169E+01	8.9146E+00	40302	2.6499E+02	2.2462E+00	2.4376E+02	2.4591E+02
82	50103	4.1166E+01	9.9309E+00	40103	2.6693E+02	2.7829E+00	2.4602E+02	2.4817E+02
83	50203	4.1165E+01	8.9309E+00	40203	2.6691E+02	2.7745E+00	2.4601E+02	2.4816E+02
84	50303	4.1165E+01	8.9309E+00	40303	2.6651E+02	2.8083E+00	2.4595E+02	2.4810E+02
85	50104	4.6085E+01	8.9796E+00	40104	2.7038E+02	3.1696E+00	2.5013E+02	2.5228E+02
86	50204	4.6085E+01	8.9795E+00	40204	2.7037E+02	3.1531E+00	2.5011E+02	2.5226E+02
87	50304	4.6082E+01	8.9794E+00	40304	2.6977E+02	3.2121E+00	2.5001E+02	2.5216E+02
88	50105	5.2802E+01	9.0642E+00	40105	2.7618E+02	3.3295E+00	2.5366E+02	2.5750E+02
89	50205	5.2801E+01	9.0642E+00	40205	2.7620E+02	3.2923E+00	2.5333E+02	2.5746E+02
90	50305	5.2790E+01	9.0637E+00	40305	2.7500E+02	3.3774E+00	2.5509E+02	2.5723E+02
91	50106	5.8267E+01	9.1671E+00	40106	2.8396E+02	3.3544E+00	2.5979E+02	2.6193E+02
92	50206	5.8264E+01	9.1670E+00	40206	2.8409E+02	3.2807E+00	2.5973E+02	2.6187E+02
93	50306	5.8238E+01	9.1659E+00	40306	2.8141E+02	3.3493E+00	2.5971E+02	2.6124E+02
94	50107	6.5042E+01	9.2798E+00	40107	2.9047E+02	3.3908E+00	2.6466E+02	2.6680E+02
95	50207	6.5038E+01	9.2797E+00	40207	2.9047E+02	3.3172E+00	2.6464E+02	2.6677E+02
96	50307	6.4979E+01	9.2776E+00	40307	2.8674E+02	3.2125E+00	2.6354E+02	2.6567E+02
97	50108	7.1751E+01	9.4054E+00	40108	2.9950E+02	3.4551E+00	2.7019E+02	2.7233E+02
								2.7683E+02
								2.7879E+02

98	50208	7.1753E+01	9.4054E+00	40208	3.0010E+02	3.4564E+00	2.7035E+02	2.7249E+02	2.7679E+02	2.7896E+02
99	50308	7.1866E+01	9.4023E+00	40308	2.9922E+02	3.0138E+00	2.6938E+02	2.7152E+02	2.7581E+02	2.7798E+02
100	50109	7.8496E+01	9.5219E+00	40109	3.1342E+02	3.5328E+00	2.7719E+02	2.7933E+02	2.8365E+02	2.8582E+02
101	50209	7.8512E+01	9.5221E+00	40209	3.1486E+02	3.5457E+00	2.7760E+02	2.7975E+02	2.8407E+02	2.8625E+02
102	50309	7.8443E+01	9.5181E+00	40309	3.2328E+02	2.8590E+00	2.7797E+02	2.8012E+02	2.8444E+02	2.8662E+02
103	50110	8.2569E+01	9.6253E+00	40110	3.4834E+02	3.5936E+00	2.8806E+02	2.9025E+02	2.9465E+02	2.9687E+02
104	50210	8.2597E+01	9.6260E+00	40210	3.5039E+02	3.6230E+00	2.8873E+02	2.9092E+02	2.9533E+02	2.9755E+02
105	50310	8.2522E+01	9.6210E+00	40310	3.6072E+02	2.9121E+00	2.8817E+02	2.9036E+02	2.9476E+02	2.9698E+02
106	50111	8.6669E+01	9.6273E+00	40111	3.8761E+02	2.6894E+00	2.9597E+02	2.9816E+02	3.0259E+02	3.0482E+02
107	50211	8.6721E+01	9.6284E+00	40211	3.8934E+02	2.8302E+00	2.9729E+02	2.9950E+02	3.0394E+02	3.0617E+02
108	50311	8.6595E+01	9.6227E+00	40311	3.9367E+02	2.2885E+00	2.9439E+02	2.9658E+02	3.0099E+02	3.0322E+02

THERMAL STRUCTURE NUMBER 5 AT TIME= 4.067E+06 SECONDS

		FLUID HEAT TRANSFER				THERMAL STRUCTURE TEMPERATURES	
ELEMENT CELL		TEMPERATURE COEFFICIENT					
142	60101	3.8403E+01	8.7402E+00	1.6875E+02	1.6875E+02		
143	60201	3.8403E+01	8.7402E+00	1.6875E+02	1.6875E+02		
144	60301	3.8403E+01	8.7402E+00	1.6875E+02	1.6875E+02		
145	60102	3.8806E+01	8.6977E+00	1.6979E+02	1.6979E+02		
146	60202	3.8806E+01	8.6977E+00	1.6979E+02	1.6979E+02		
147	60302	3.8806E+01	8.6977E+00	1.6979E+02	1.6979E+02		
148	60103	4.0175E+01	8.7009E+00	1.7111E+02	1.7111E+02		
149	60203	4.0175E+01	8.7009E+00	1.7111E+02	1.7111E+02		
150	60303	4.0175E+01	8.7009E+00	1.7111E+02	1.7111E+02		
151	60104	4.3567E+01	8.7068E+00	1.7441E+02	1.7441E+02		
152	60204	4.3567E+01	8.7068E+00	1.7441E+02	1.7441E+02		
153	60304	4.3567E+01	8.7068E+00	1.7441E+02	1.7441E+02		
154	60105	4.8252E+01	8.7206E+00	1.7889E+02	1.7889E+02		
155	60205	4.8252E+01	8.7206E+00	1.7889E+02	1.7889E+02		
156	60305	4.8252E+01	8.7206E+00	1.7889E+02	1.7889E+02		
157	60106	5.2126E+01	8.7274E+00	1.8266E+02	1.8266E+02		
158	60206	5.2126E+01	8.7274E+00	1.8266E+02	1.8266E+02		
159	60306	5.2126E+01	8.7273E+00	1.8267E+02	1.8267E+02		
160	60107	5.7041E+01	8.7202E+00	1.8769E+02	1.8769E+02		
161	60207	5.7041E+01	8.7202E+00	1.8769E+02	1.8769E+02		
162	60307	5.7041E+01	8.7201E+00	1.8769E+02	1.8769E+02		
163	60108	6.2023E+01	8.7134E+00	1.9277E+02	1.9277E+02		
164	60208	6.2023E+01	8.7135E+00	1.9277E+02	1.9277E+02		
165	60308	6.2023E+01	8.7132E+00	1.9277E+02	1.9277E+02		
166	60109	6.7080E+01	8.7152E+00	1.9780E+02	1.9780E+02		
167	60209	6.7080E+01	8.7152E+00	1.9780E+02	1.9780E+02		
168	60309	6.7080E+01	8.7147E+00	1.9781E+02	1.9781E+02		
169	60110	7.0119E+01	8.7108E+00	2.0091E+02	2.0091E+02		
170	60210	7.0119E+01	8.7109E+00	2.0090E+02	2.0090E+02		
171	60310	7.0120E+01	8.7104E+00	2.0091E+02	2.0091E+02		
172	60111	7.3202E+01	8.7603E+00	2.0325E+02	2.0325E+02		
173	60211	7.3202E+01	8.7603E+00	2.0325E+02	2.0325E+02		
174	60311	7.3203E+01	8.7599E+00	2.0326E+02	2.0326E+02		

E.4 Graphics Output

The binary-format restart and plot files written by the COMMIX-1AR/P computer program during execution of the sample problem have been used as input to the four postprocessor programs described in Appendix D to produce several forms of graphical output. The results of these postprocessing operations are shown in this section.

The VECT computer program was used to produce two graphical displays of the velocity distribution at the end of the steady-state calculation. The input provided to the VECT program consisted of the binary-format restart file written by the COMMIX-1AR/P program and the card-image input file shown in Table D.3; the second paragraph of Sec. E.2.1 describes how cells are arranged in the planar views produced by the VECT program. Figure E.4 shows the velocity distribution in J plane numbers 1 (on the right) and 3 (on the left) using vectors to represent both the magnitude and the direction; the reference vector at the bottom of the figure shows the absolute velocity scale. The sodium flow within the reactor vessel (RV) is up through the cold pool, fuel assemblies, and hot pool (I=1), down through the intermediate heat exchanger (IHX) (I=2 and J=3), up through the pump (I=2 and J=2, which is not shown in the figure), and down through the pipe (I=2 and J=1). There is a bouyancy-driven natural circulation of argon upward along the outside of the RV (I=3) and downward along the inside of the guard vessel (GV) (I=4). Air flow is upward in the gap between the GV and the cavity vessel (CV) (I=5 and 6). Sodium flow is upward on the intermediate side of the IHX (I=8 and J=3). Figure E.5 shows the velocity distribution in K plane number 10. This figure shows the sodium in the RV moving toward the IHX inlet (which is just below this K plane at I=2 and J=3).

The TEMP computer program was used to produce two graphical displays of the temperature distribution at the end of the steady-state calculation. The input provided to the TEMP program consisted of the binary-format restart file written by the COMMIX-1AR/P program and the card-image input file shown in Table D.5; the arrangement of cells in the planar views produced by the TEMP program is identical to the arrangement produced by the VECT program. Figure E.6 shows the temperature distribution in J plane numbers 1 (on the

$J = 3, 1$ $\text{TIME} = 4.07 \cdot 10^6 \text{ sec.}$
 $\longrightarrow 9.19 \text{ m/s (} 30.15 \text{ ft/s)}$

Fig. E.4 Steady-State Velocity Distribution for Sample Problem
 for Planes $J=3$ (left) and $J=1$ (right).

$K = 10$ $\text{TIME} = 4.07 \cdot 10^6 \text{ sec.}$
 $\longrightarrow 7.12 \cdot 10^{-2} \text{ m/s } (2.34 \cdot 10^{-1} \text{ ft/s})$

Fig. E.5 Steady-State Velocity Distribution for Sample Problem for Plane $K=10$.

right) and 3 (on the left) using dashed-line isotherms; the minimum and maximum temperatures are indicated. Isotherms are drawn by performing a multidirectional interpolation between the temperatures at adjacent cell centers (including effects of boundary surfaces, if present); therefore, the lines showing the increase in temperature through the fuel assembly region ($I=1$) are present between the centers of cells at $K=4$ and those at $K=5$, even though all heat is added within the $K=5$ level. Similarly, the decrease in temperature on the primary side of the IHX ($I=2$ and $J=3$) is shown between the middle of cells at $K=10$ and those at $K=7$, even though all heat is removed within levels $K=9$ through $K=7$. The absence of isotherms in the remainder of the RV interior indicates the presence of two rather uniform temperature regions: the hotter region (upper) is between 490 and 499 °C and the colder region (lower) is between 360 and 370 °C. (The closely clustered isotherms at cell (2,3,10) are an artifice of the program performing an interpolation between the temperature at the center of this fluid cell and the temperature of the boundary surfaces which separate $I=1$ from $I=2$ at $K=9$.) The gap between the RV and the GV ($I=3$ and 4) is too narrow for the isotherms to be distinguishable; however, the close clustering of isotherms indicates a significant temperature change across the RV/GV gap. The gap between the GV and the CV ($I=5$ and $I=6$) shows a temperature increase from bottom to top and a temperature decrease at increasing radial position. (Interpretation of isotherm figures is usually aided by referring to the printable output file from the COMMIX run or by additionally drawing a digit-format display (IDIGIT $\neq$ 0) of the same plane view using the TEMP program.) Figure E.7 shows the temperature distribution in K plane number 9 using the digit-format display. This cross-sectional view shows the sodium temperature in the RV is hot in the central pool ($I=1$), cooler in the IHX ($I=2$ and $J=3$), and coolest in the pump ($I=2$ and $J=2$) and the pipe ($I=2$ and $J=1$). There are both radial and azimuthal temperature gradients in the argon in the RV/GV gap ($I=3$ and $I=4$). There is a radial (but not azimuthal) temperature gradient in the air in the GV/CV gap ($I=5$ and $I=6$).

Fig. E.6 Steady-State Temperature Distribution (Isotherm Spacing of 10°C) for Sample Problem for Planes $J=3$ (left) and $J=1$ (right).

$K = 9$ $\text{TIME} = 4.07 \cdot 10^6 \text{ sec.}$
 TEMPERATURE (C)

Fig. E.7 Steady-State Temperature Distribution for Sample Problem for Plane K=9.

The GRPH computer program was used to produce six graphical displays of the manner in which variables were changing during the steady-state calculation. In preparation for using the GRPH program, the TDSC program was used to extract a subset of data from the binary-format plot file written by the COMMIX-1AR/P program; the card-image input supplied to the TDSC program is shown in Table D.7b; data were extracted for three velocity components and seven temperatures. The input provided to the GRPH program consisted of the binary-format file written by the TDSC program and the card-image input file shown in Table D.9; the time range was restricted to the first 10^4 s and, thus, the figures only show the initial 220 time steps during the approach to steady state; data values are available for every tenth time step, and these data values are connected by straight lines (which results in the curves being somewhat discontinuous in slope at each data value). Figure E.8a shows the axial velocity at the +K face of fluid cell location (2,2,7), which is in the pump; the bulk flow rate of sodium in the RV has stabilized by 10^3 s. Figure E.8b shows the axial velocity at the +K face of two fluid cells containing argon at mid-height in the RV/GV gap: the solid line is for cell (3,2,5), which is adjacent to the RV, and the dotted line is for cell (4,2,5), which is adjacent to the GV; these two velocities are still changing slowly after 10^4 s; however, the general pattern of flow upward adjacent to the (hotter) RV and downward adjacent to the (colder) GV was established much earlier in the calculation. Figure E.9a shows the temperature for two sodium cells within the RV: the solid line is for cell (1,2,4), which is at the core inlet, and the dotted line is for cell (1,2,5), which is at the core outlet; the difference between these two temperatures has stabilized by 10^3 s (corresponding to stabilization of the bulk core flow rate); however, both temperatures are increasing slowly during the remainder of the history shown (as the system approaches equilibrium). Figure E.9b shows the temperature for fluid cell (8,3,10), which is the exit of the intermediate side of the IHX; this temperature is increasing at a slow rate after 3×10^3 s. Figure E.9c shows the argon temperature adjacent to the RV at two axial locations: the solid line is for cell (3,2,1), which is at the bottom of the RV/GV gap, and the dotted line is for cell (3,2,11), which is at the top of the RV/GV gap; both of these temperature are reasonably stable after 2×10^3 s. Figure E.9d shows the air temperature at the top of the GV/CV gap for two radial locations: the solid line is for cell (5,2,11), which is adjacent to the GV, and the dotted line is for cell (6,2,11), which is adjacent to the CV; these two temperatures and the difference between these two temperature are all continuing to change during the history shown, indicating

that the rate at which heat is removed by the flowing air is continuing to change. Taken together, the results shown in Figs. E.8 and E.9 show (1) the solution is very close to steady state after 2×10^3 s (about 200 time steps), (2) the conditions inside the RV are converging sooner than the conditions outside the RV, and (3) of the quantities shown, the temperature of the air in the GV/CV gap is slowest to converge.

Fig. E.8 Velocity Histories During Approach to Steady State for Sample Problem.

(a) Core In/Out: TL(1,2,4), TL(1,2,5)

(b) IHX-int Exit: TL(8,3,10)

(c) Argon Bot/Top: TL(3,2,1), TL(3,2,11)

(d) Air Exit: TL(5,2,11), TL(6,2,11)

Fig. E.9 Temperature Histories During Approach to Steady State for Sample Problem.

E.5 Continuations

The user may find it necessary to perform a desired calculational sequence as a series of computer runs. The majority of the input needed in the continuation runs is provided by the binary-format restart file, which is written by a prior run and read by a later run. The user need only supply minimal input in the card-image input file for the later runs. Shown below are three examples; the general rule is to only specify what is essential.

An example of the card-image input file used to continue a steady-state calculation is shown in Table E.4. The use of both an input and an output restart file is selected by IFRES=3 in NAMELIST GEOM. The continuation of a steady-state calculation is indicated by ISTATE=1 in NAMELIST DATA. The user may want (or need) to specify revised input for variables associated with case control (e.g. NTMAX, TIMAX, EPSi, and NTHCON) and output (e.g. NTPRNT, NTHPR, xxxBUG, and NTPLOT). The two "END " records at the end of the file mark the ends of the Boundary Value Initialization and Internal Cell Initialization input blocks; even though no values are to be changed using these two blocks, the "END " terminator records must always be present.

Table E.4 Card-Image Input File to Continue a Steady-State Calculation

&GEOM	IFRES=3,	&END
&DATA		
	ISTATE=1,	
	&END	
END		BNDRY. VALUE INIT.
END		INTERNAL CELL INIT.

An example of the card-image input file used to begin a transient calculation from a steady-state calculation is shown in Table E.5. The beginning of a transient calculation is indicated by ISTATE=2 in NAMELIST DATA. At the beginning of a transient, the user must reset the initial case time (TSTART) to zero and allow more than one outer iteration per time step (array IT). Any transient functions to be used must be defined (using arrays TVAL, FVAL, and NEND); the sample problem in Sec. E.2 allowed for two such functions; the input

values specified here are the examples shown in Sec. 4.1.4. The user will usually need to respecify the initial time step size (using array DT, to override the value being used at the end of the steady-state calculation), the maximum problem time (TIMAX), and the time step numbers at which long edits are to be printed. If the case uses radiation heat transfer surfaces, the small value of the relaxation factor (OMR) used during the steady-state calculation should usually be overridden by the user with a larger value for the transient calculation. If the case uses the pump model, NAMELIST PUMP and the Pump Location input block must be present in the card-image input file even if no new values are to be specified.

Table E.5 Card-Image Input File to Begin a Transient Calculation
from a Steady-State Calculation

```

&GEOM  IFRES=3,  &END
&DATA
  ISTATE=2, TSTART=0.0, IT=2*30,
  TIMAX=5.0, DT=0.01,0.01, OMR=0.8,
  NTPRNT=-10,0,
  TVAL=0.,1.,2.,3.,2*4.,5., 0.,2*1.,2*1.1,5.,
  FVAL=1.,1.3,1.,.7,3*1., 3*1.,3*1.,
  NEND=7, 6, 0,
&END
&PUMP  &END
END          PUMP LOC.
END          BNDRY. VALUE INIT.
END          INTERNAL CELL INIT.

```

An example of the card-image input file used to continue a transient calculation is shown in Table E.6. The continuation of a transient calculation is indicated by ISTATE=3 in NAMELIST DATA. The user may want (or need) to specify revised input for the same types of items (e.g. case control and output) as were mentioned for continuing a steady-state calculation (ISTATE=1).

Table E.6 Card-Image Input File to Continue a Transient Calculation

```

&GEOM  IFRES=3,  &END
&DATA
  ISTATE=3,
&END
END          BNDRY. VALUE INIT.
END          INTERNAL CELL INIT.

```

COMMIX-1AR/P Computer Program Problem or Suggestion Report (9/23/91)

Complete the top part of this form and submit to one of the persons whose name is shown on the banner page of the Input Guide or of your output file.

Submitted by: Name: _____ Date: __/__/__

Mailing Address: _____

Telephone Number: ____-____-____ E-mail Address: _____

Type of Problem: Execution: __ Documentation: __ Suggestion: __ Other: _____

Code Version (from banner page): Name: _____ Date: __/__/__

Machine: Brand: _____ Model: _____ Operating System: _____

Description of Problem or Suggestion (be as specific as possible):

To be completed by COMMIX-1AR/P Code Maintenance Staff

Received: __/__/__ Source: phone: __ mail: __ E-mail: __ FAX: __ other: _____

Log Number: _____ Priority: _____ Assigned to: _____

Action(s) (with date and by whom):

Distribution for ANL-92/33

Internal:

C. H. Adams	P. L. Garner (25)	R. M. Singer
R. N. Blomquist	E. M. Gelbard	M. J. Tan
M. Bottoni	K. C. Gross	A. M. Tentner
L. L. Briggs	D. J. Hill	C. E. Till
F. B. Brown	Kalimullah	C. P. Tzanos
J. E. Cahalan	H. S. Khalil	R. A. Valentin
T. R. Canfield	M. Minkoff	R. B. Vilim
Y. S. Cha	D. Mohr	D. C. Wade
Y. I. Chang	E. E. Morris	D. P. Weber
T. H. Chien	H. M. Park	T. Y. Wei
D. H. Cho	P. A. Pizzica	R. A. Wigeland
H. M. Domanus	H. P. Planchon	B. S. Yarlagadda
F. E. Dunn	R. C. Schmitt	ANL Patent Department
E. E. Feldman	W. T. Sha	ANL Contract File
P. J. Finck	Y. W. Shin	TIS Files (3)
E. K. Fujita	J. J. Sienicki	

External:

DOE-OSTI for distribution per UC-542 (68)
ANL-TIS Libraries
Chicago Operations Office, DOE:
 A. L. Taboas
 J. L. Hooper
DOE Headquarters, Washington, DC:
 H. Alter
 P. B. Hemmig
 J. W. Herczeg
J. M. Berkoe, Bechtel Power Corporation, San Francisco, CA
P. Kroeger, Brookhaven National Laboratory
G. Van Tyle, Brookhaven National Laboratory
D. Grand, CEA-Centre d'Etudes Nucleaires de Grenoble, France
M. Parent, CEA-Centre d'Etudes Nucleaires de Grenoble, France
T. Charlton, EG&G Idaho Inc., Idaho Falls
S. Z. Rouhani, EG&G Idaho Inc., Idaho Falls
E. Brega, ENEL-Thermal and Nuclear Research Center, Milan, Italy
M. McDowell, Energy Technology Engineering Center
A. Shenoy, General Atomics, San Diego, CA
P. MaGee, General Electric Company, San Jose, CA
J. A. Turner, Los Alamos National Laboratory
N. E. Todreas, Massachusetts Institute of Technology, Cambridge
M. Oszewski, Oak Ridge National Laboratory
S. Aslam, University of Illinois, Urbana
S. Gallopoulos, University of Illinois, Urbana
M. Berzins, University of Leeds, England
N. Aslam, University of Michigan, Ann Arbor
J. Dorning, University of Virginia, Charlottesville
Rizwan-uddin, University of Virginia, Charlottesville

**DATE
FILMED**

3 / 17 / 93