K/CSD/TM-3

COMPUTER SCIENCES DIVISION

HEXEREI2: COMPUTER PROGRAM FOR THE TRANSIENT
THERMAL-HYDRAULIC ANALYSIS OF HIGH TEMPERATURE
GAS-COOLED REACTORS

G. E. Giles
W. D. Turner
K. W. Childs
R. M. DeVault
B. R. Becker

UNION
CARBIDE

OAK RIDGE GASEOUS DIFFUSION PLANT
OAK RIDGE, TENNESSEE

prepared for the U.S. ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
under U.S. GOVERNMENT Contract W-7405 eng 26

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

48
8-30-77
25 copy to NTIS
21
ACRS Committee

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy \$6.75; Microfiche \$3.00

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISCLAIMER

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

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Contract No. W-7405 eng 26

COMPUTER SCIENCES DIVISION

HEXEREI2: Computer Program for the Transient Thermal-
Hydraulic Analysis of High Temperature
Gas-Cooled Reactors

G. E. Giles
W. D. Turner
K. W. Childs
R. M. DeVault
B. R. Becker

May, 1977

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

NOTICE

This document contains information of a preliminary nature. It is subject to revision or correction and therefore does not represent a final report.

UNION CARBIDE CORPORATION, NUCLEAR DIVISION

operating the

Oak Ridge Gaseous Diffusion Plant

Oak Ridge Y-12 Plant

Oak Ridge National Laboratory

Paducah Gaseous Diffusion Plant

for the

ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

LIST OF ILLUSTRATIONS (Contd.)

<i>Figure</i>		<i>Page</i>
A-3.	Format of JCL to Run HEXEREI2 at ORGDP when FORTRAN Deck is Modified.	144
C-1.	Subroutine BNDTMP.	150
C-2.	Subroutine HEATGN.	151

CONTENTS

	<i>Page</i>
List of Illustrations	v
List of Tables	vii
Abstract	1
1. General Introduction	3
2. General Description of HEXEREI2	9
2.1. Capabilities of the HEXEREI2 Code	9
2.2. Nodal Mesh Generation	10
2.3. Mathematical Description	16
2.3.1. Statement of the Problem	16
2.3.2. Steady-State Heat Conduction	20
2.3.3. Transient Heat Conduction	22
2.3.3.1. Classical Explicit Procedure	23
2.3.3.2. Levy's Modification to Classical Explicit Procedure	23
2.3.3.3. Implicit Transient Numerical Procedure	24
2.3.3.4. Variable Time Step Size	29
2.3.4. Temperature-Dependent Thermal Properties	30
2.3.5. Boundary Conditions	31
2.3.6. Calculation of Coolant Temperatures	32
2.3.7. Flow Redistribution Technique for Transient Analysis	40
2.3.8. Flow Redistribution for Steady-State Analysis	46
2.3.9. Modeling Technique for HTGR Auxiliary Heat Exchangers	47
2.3.9.1. General	47
2.3.9.2. Fort St. Vrain Steam Generator	55
2.3.10. CACS Circulator Simulation	55
2.3.11. Coolant Properties	60
3. Description of Input and Output	63
3.1. General	63
3.2. Regions	63
3.3. Lattice Arrangement	63
3.4. Analytical and Tabular Functions	64
3.5. Input Parameters	64
3.6. Card Description	66
3.6.1. Card 1	66
3.6.2. Card 2	66
3.6.3. Card 3	67
3.6.4. Card 4	69
3.6.5. Card 5	70
3.6.6. Region Data	73
3.6.7. Material Data	74

CONTENTS (Contd.)

	<i>Page</i>
3. Description of Input and Output (Contd.)	
3.6.8. Heat Generation Function Data	75
3.6.9. Initial Temperature Function Data	75
3.6.10. Boundary Data	75
3.6.11. Lattice Description	77
3.6.12. Analytical Function Data	78
3.6.13. Tabular Function Data	79
3.6.14. Output Times	80
3.6.15. Node Numbers for Special Monitoring of Temperatures	80
3.6.16. HTGR Input Block	80
3.6.17. Initial Temperatures	82
3.6.18. Implicit Transient Technique Parameters	83
3.6.19. Blank Card	85
3.7. Problem Size Limits	85
3.8. Summary and Format of Input	85
3.9. Output Description	85
3.9.1. General	85
3.9.2. Input Return	85
3.9.3. Temperature Map	98
3.9.4. Summary Information	98
3.9.5. Steady-State Temperature Distribution	98
3.9.6. Transient Temperature Distribution	98
3.9.7. Special Monitoring of Temperatures	98
4. HEXEREI2 Evaluation	111
4.1. Three-Dimensional Conduction	111
4.2. Boundary Heat Transfer	116
4.3. Coupled Convection to Flowing Coolant	116
5. Discussion of HEXEREI2 Analyses as a User Aid	125
5.1. Sample Input for General Test Cases	125
5.2. Fort St. Vrain Analysis	125
5.3. Time Step Size Consideration	135
Acknowledgments	137
References	139
Appendix A. Control Cards	141
Appendix B. Nomenclature	145
Appendix C. User-Supplied Subroutines	149

LIST OF ILLUSTRATIONS

<i>Figure</i>	<i>Page</i>
1.1. Fort St. Vrain HTGR General View.....	4
1.2. GAC's 2000/3000 Megawatt HTGR General View	5
2.2.1. HTGR Core Plan.....	11
2.2.2. HEXEREI Typical Cell Geometry for One Axial Plane	12
2.2.3. Typical Hexagonal Cell with Sample Connector from Node 1 to Node 2	13
2.2.4. Typical Interface Cells with Sample Connectors.....	14
2.2.5. Possible Cell Distribution for a Reduced Number of Nodes	15
2.2.6. Comparison of GAC and HEXEREI2 Core Maps	17
2.3.1.1. Typical Solid Node	18
2.3.6.1. Coolant and Solid Cells	33
2.3.6.2.a Regular Fuel Elements	34
2.3.6.2.b Control Rod Fuel Elements	34
2.3.6.3. Graphical Representation of Coolant Temperature Solution	36
2.3.6.4. Coolant Channel Description with Sample Node Locations.....	38
2.3.6.5. Sample Coolant Channel Modification in Lower Reflector and Core Support Block	39
2.3.9.1. Core Auxiliary Cooling Loop Arrangement	48
2.3.9.2. Heat Exchanger Tubing Distribution in R-Z Plane	49
2.3.9.3. Graphical Representation of Heat Exchanger Calculations	51
2.3.9.4. Heat Exchanger Model with Sample Sections	52
2.3.9.5. Fort St. Vrain Steam Generator – Simplified Description.....	56
2.3.10.1. Sample Circulator Performance Map	59
3.6. Plot of a Sample Tabular Function	79
3.9.2.1. Input Return from HEXEREI2.....	89
3.9.4.1. Summary Output from HEXEREI2	99
4.1. Model for General Test Cases	112
4.2. Comparison of Radial Temperature Distribution Test Case Number 1 (Uniform Heat Generation, Constant Properties, Constant Boundary Temperatures)	113
4.3. Comparison of Radial Temperature Distribution Test Case Number 2 (Uniform Heat Generation, Nonlinear Properties, Constant Boundary Temperatures).....	114

LIST OF ILLUSTRATIONS (Contd.)

<i>Figure</i>		<i>Page</i>
4.4.	Comparison of Radial Temperature Distribution for Transient Test Case with Constant Properties and Boundary Temperature and No Heat Generation.....	115
4.5.	Steady-State Axial Temperature Distribution for Test Case with Boundary Condition Applied to Both Ends of Model	117
4.6.	Steady-State Radial Temperature Distribution for Test Case with Boundary Condition Applied to Outer Radius.....	118
4.7.	Analogous Models for HEXEREI and CCCM5.....	119
4.8.	Comparison of Axial Coolant Temperature Distribution. Test Case Number 7 (Transient, Uniform and Constant Properties, Heat Generation and Coolant Flow Rate)	120
4.9.	Comparison of Axial Solid Temperature Distribution. Test Case Number 7 (Transient, Uniform and Constant Properties, Heat Generation, and Coolant Flow Rate). CCCM Points Are Volume Temperatures	121
4.10.	Comparison of Axial Coolant Temperature Distribution. (Transient, Uniform and Constant Properties, Heat Generation and Coolant Flow Rate).....	122
4.11.	Comparison of Axial Solid Temperature Distribution. Test Case Number 7 (Transient, Uniform, and Constant Properties, Heat Generation, and Coolant Flow Rate). CCCM Points Are Volume Average Temperatures	123
5.1.1.	Data for Test Case Number 1	126
5.1.2.	Data for Test Case Number 5	126
5.1.3.	Data for Test Case Number 7	127
5.2.1.	Geometric Model for HEXEREI2 Analysis of Fort St. Vrain HTGR	128
5.2.2.	Data for HEXEREI2 Analysis of Fort St. Vrain HTGR	129
5.2.3.	FORTTRAN Listing of User-Supplied Subroutine CPHEAT	131
5.2.4.	FORTTRAN Listing of User-Supplied Subroutine CONDTN Demonstrating Anisotropic Conductivity Modeling.....	132
5.2.5.	Region Power Peaking Factors	133
5.2.6.	FORTTRAN Listing of User-Supplied Subroutine HEATGN	134
A-1.	Format of JCL to Run HEXEREI2 at ORGDP.....	141
A-2.	Main Program for HEXEREI2 100 Nodes	143

LIST OF TABLES

<i>Table</i>	<i>Page</i>
2.1 Logic to Determine Whether ω Should be Increased or Decreased	29
2.2 Steam Generator and Reheater Physical Characteristics	57
3.1 Dependence of Input Parameters	65
3.2 A Cards Necessary to Describe the Function 100 sin5x + exp(-3x)	79
3.3 T Cards Necessary to Describe the Sample Tabular Function Depicted in Fig. 3.6	80
3.4 Problem Size Limits	86
3.5 Summary of Input	87
4.1 Material Properties	111

ABSTRACT

HEXEREI2 code development has provided a complex computer code, which is capable of analyzing High Temperature Gas-Cooled Reactors (HTGR) during accidents and providing most of the information necessary for HTGR Safety Studies.

HEXEREI2 is designed to solve steady-state and transient three-dimensional heat conduction, coupled with convection to axially flowing coolant in a large number of parallel channels. Both the coolant temperature and flow distribution among the channels are influenced by the heat transferred into coolant. The thermal conductivity, density, and specific heat may be spatially and temperature dependent and may further be modified by HTGR design parameters. The conductivity can be anisotropic. Heat generation rates may be dependent on time, position, and HTGR design parameters. The boundary temperatures may be time-dependent. The boundary conditions may be fixed temperatures or any combination of prescribed heat flux, forced convection, and radiation from a surface to a boundary temperature. The boundary condition parameters may be time- and/or temperature-dependent. HEXEREI2 can model a coolant which can be helium, air, or a helium-air mixture. The nodal mesh is automatically generated from a few simple input parameters and produces a model capable of an accurate thermal-hydraulic analysis of an HTGR with present computers. The mesh generated emulates the HTGR in geometric structure and the code can analyze the largest HTGR design proposed to date.

The maximum number of nodes can be easily adjusted to fit the problem and computer storage limitations. The core requirement on an IBM 360 machine ranges from 314K bytes for one node to 1500K bytes for 6000 nodes.

HEXEREI2 has the ability to model heat removal from a closed system by a water-gas heat exchanger accepting the exit gas from the core and returning the heat exchanger's exit gas to the core inlet. With the circulator model in the code, it is possible to conduct completely self-sufficient transient analysis on the HTGR core and cooling system.

The point successive overrelaxation iterative method and a modification of the "Aitken δ^2 extrapolation process" are used to solve the finite difference equations which approximate the partial differential equations for a steady-state problem.

The transient problem may be solved using any one of several finite difference schemes. These include an implicit technique which can range from Crank-Nicolson to the Classical Implicit Procedure, an explicit method which is stable for a time step of any size, and the Classical Explicit Procedure which involves the first forward time difference. The solution of the system of equations arising from the implicit technique is accomplished by point successive overrelaxation iteration, and includes procedures to estimate the optimum acceleration parameter.

▼

1

1

1

1

1. GENERAL INTRODUCTION

This report is intended to provide future users with enough information to utilize the HEXEREI2 computer code. In addition, the results of studies and comparisons are presented to document HEXEREI2's accuracy. It is hoped that a frank discussion of the performance of the code will prevent future users from using the code as a 'black box', but rather use the code in the most accurate manner.

HEXEREI2 is a three-dimensional coupled conduction-convection heat transfer and multichannel fluid dynamic analysis computer code for both steady-state and transient problems. The program was developed to provide thermal-fluid dynamic analysis of High Temperature Gas-Cooled Reactors (HTGR) to satisfy licensing procedures for the Nuclear Regulatory Commission (NRC).

HEXEREI2's development was commissioned by the Nuclear Regulatory Commission (NRC) to provide an analysis capability independent of those used by the reactor manufacturers and which was not available at Oak Ridge or other sites. The General Atomic Corporation (GAC) is the only active HTGR manufacturer in the United States and their designs represent a departure from both European gas cooled reactors and normal light water reactors. HEXEREI2's development was restricted somewhat to GAC's designs and was initiated in June 1974 with the goal of producing a code that could provide a safety analysis capability. Improvements on GAC's RECA technique [Ref. 1] were possible since the project was starting with a previously developed code (HEATING5 [Ref. 2]) and had the benefit of recognizing the deficiencies of existing codes. Another major goal was to develop an exportable code, so that NRC personnel with different computer systems could use the HEXEREI2 code to develop expertise and confidence in the safety analysis of HTGRs. The development effort has been level at about one and one-half man years since December 1974. Since HEXEREI2 was developed from HEATING5, much of the report is a modification of the HEATING5 report [Ref. 2].

The HEXEREI2 computer code provides the steady-state, transient three-dimensional heat conduction analysis for an automatically generated mesh with a central portion that is a regular hexagonal mesh in the axial plane. This analysis technique is coupled with convection to coolant flowing in a system of parallel axial channels. The coolant temperatures are solved from the coupled thermal-fluid dynamic analysis.

HEXEREI2 contains a very general mathematical package for the solution of the solid and coolant temperatures. The user has the capability of selecting a steady-state technique, an explicit transient technique normally too expensive for the general three-dimensional problem, or a very powerful implicit transient technique. The implicit transient technique is capable of reducing the computing cost of an analysis considerably. However, for problems requiring very small time steps, the explicit technique compares more favorably in expense than the implicit. The treatment of the boundary condition, material properties, and heat generation rate is very general in HEXEREI2 and allows the user great flexibility in modeling. The hexagonal mesh generator, coolant temperature and flow redistribution techniques, coolant channel description and coolant system are somewhat limited to HTGR type problems, however. HEXEREI2 can analyze problems other than heat transfer in an HTGR, but since this was not within the scope of the developmental project, modifications to the code may be required.

GAC has designed and built one HTGR at Fort St. Vrain, Colorado, to be a demonstration plant producing up to 330 megawatts of electricity (Fig. 1.1). This reactor is one of a kind and has many design features that differ considerably from GAC's more recent designs (Fig. 1.2). GAC's recent designs feature 2000 megawatt and 3000 megawatt thermal HTGRs of very similar construction. The differences will be discussed later in this report if they impacted upon the development of HEXEREI2. Unless the item under discussion is identified as belonging to a specific reactor, it is assumed generic to all three GAC designs for HEXEREI2 analysis.

Fig. 1.1. Fort St. Vrain HTGR General View
(Ref. 3, Fig. 3.1-1)

ORNL-DWG 76-10928

3 RS 72091B
7/73

Fig. 1.2. GAC's 2000/3000 Megawatt HTGR General View
(Ref. 4, Fig. 4.1.3-1)

The HTGR's designed by GAC are all of the same general family. They all use helium as their primary coolant, graphite as their core support structure and moderator, and are enclosed in prestressed concrete reactor vessels. The reactor cores are each an assembly of hexagonal graphite prisms arranged in a right circular cylinder (Fig. 1.1). The fuel is contained in blind axial holes in the fuel elements (Fig. 1.1). Helium coolant flow is through many axial channels in the fuel elements. The fuel elements are arranged in rings of six columns of regular fuel elements around a single column of control rod fuel elements. This arrangement of seven columns is called a refueling region and all of the fuel in this region has approximately the same region power peaking factor or heat generation rate. Since all seven columns of fuel elements in a refueling region are generally reloaded at the same time, the amount of irradiation of the graphite and the age of the fuel in one refueling region are uniform.

The active core is surrounded by graphite blocks arranged as a reflector mass which contains no fuel but may allow coolant flow. Helium flows: down through the fuel element coolant channels, the cracks around the fuel elements, control rods, and side reflector elements; into the lower plenum; through cross flow ducts; up through the Core Auxiliary Cooling System (CACS) heat exchanger bundle; through the CACS circulator; and into the upper plenum to return to the core (Fig. 1.2).

The CACS heat exchanger and main steam generator are helically coiled water tube heat exchangers. Both the main and CACS circulators are single stage axial compressors with a steam turbine driving the main and an electric motor driving the CACS circulator.

The Fort St. Vrain reactor has no CACS but, instead, uses the main steam generators with alternative water supplies and the main circulator with an alternative water turbine drive to supply emergency cooling. Analysis of Fort St. Vrain reactor systems was included in HEXERE12's analysis capabilities.

Heat transfer in the HTGR core involves both heat conduction through the graphite and convection to the axially flowing coolant in the myriad of coolant channels. The coolant flow during high flow rates controls the heat transfer and will tend to produce in each refueling region temperatures that are uniform in the axial plane. This is due to the refueling region's uniformity of heat generation, the uniform distribution of a large number of small coolant channels which will have approximately the same coolant mass flow rate, and the high mass flow rate which tend to reduce the thermal gradients in the solid and axial plane and control the heat transfer in the axial direction.

As the mass flow rate changes, the relative retarding effects in each channel due to bouyancy, friction, acceleration, expansion, and contraction losses change. The mass flow through the core may redistribute itself among the coolant channels since the change in retarding forces is not uniform throughout the core. Initial temperature, mass flow rates and orifice settings differences contribute to the nonuniformity of change in retarding forces. If the flow were to be reduced to near zero, the bouyant forces produced in the hot channel could retard the flow to stagnation and perhaps drive the flow upwards in the channel (reverse flow). Also at the lower flow rates, heat conduction becomes more important. When the net flow through the core is stopped, natural convection loops can develop in the core and over long periods will redistribute the heat until the temperatures are uniform throughout the core. During an accident transient, the coolant flow rate may change from full flow to zero or any level in between. During full flow, conductive heat transferred is very small relative to the convective heat transfer. As the flow decreases, conductive heat transfer becomes more important. Also, reversed (up) flow becomes more possible as the flow rate decreases. Thus, the controlling portion of the physical system may change with time. Accurate heat conduction, heat convection, and fluid dynamic models for all possible conditions that may occur during a transient are requirements for any code that is to be utilized on HTGR accident analysis.

During the safety studies of GAC's HTGR designs, various ORNL computer codes have been evaluated. One of the first utilized at Oak Ridge for HTGR analysis was HEATING5 (or

HEATING4, an earlier version). HEATING5 is a very general, one-, two-, or three-dimensional (either Cartesian or cylindrical coordinate system) heat conduction analysis technique for steady-state or transient problems. General mathematical techniques, boundary conditions, and material properties make HEATING5 a very versatile code. However, HEATING5 requires all model boundaries (external and internal) to be parallel to the coordinate axes. Thus, an HTGR core would have to be modeled in Cartesian or cylindrical coordinates. It was found from experience with HEATING5 and from GAC's code development experience that the finest Cartesian or cylindrical grid structure able to fit on present available computers could not perform an adequate analysis, due to lumping several refueling regions into one cell. Since the thermal properties, age of the fuel and power are fairly uniform within the refueling regions, it would be desirable to have at least one column of nodes per refueling region. If this criterion is applied to HEATING5 mesh generation, the refueling regions near the core center have many nodes per region. This surplus of nodes is more than annoying, since the small spacing reduces the maximum time step that can be used in the transient analysis. The large number of nodes also increases the computational work per time step and core storage requirement beyond current computer capacity. HEATING5 also lacks the ability to model the flow in a multichannel system or to couple the conduction solution to the convection solution so that the convecting fluid's temperature is a function of the heat transferred into the fluid.

CCCM5 [Ref. 5] was developed from HEATING5 to provide the ability to analyze the steady-state and transient conduction in two or three dimensions coupled with convection to axially flowing coolant along the outer boundary of the model. The coolant temperature is thereby modified by the heat flow into it. Single channel analysis of the HTGR is thus available on CCCM5 and has been extensively performed [Ref. 6]. Multichannel coolant flow, an accurate model description of the whole core able to fit in present computers, and the analysis of CACS coupled with the core are still not possible with CCCM5. The analysis capability of CCCM5 is similar to the single average channel capability of GAC's TAP [Ref. 7]. TAP can also include the whole primary and secondary coolant systems in its analysis technique, however.

The single channel analysis of the HTGR core with TAP or CCCM5 neglects the effects of interregional heat flow and coolant flow redistribution within the multitude of axial channels due to high temperature gradients and low flow rates. These two effects become very important during accidents involving reactor trip, loss of primary coolant pressure, or loss of primary coolant flow with restart of flow using the CACS. GAC uses a complex system of routines called RECA to analyze these and other accidents. RECA uses a nodal system which, in the axial plane, consists of one node for each refueling region and one or more nodes in the side reflector. Each node is then connected (heat is allowed to flow) to each of six neighboring nodes (refueling regions) in the same axial plane and one node in each of the adjacent axial planes. This nodal system provides a sufficiently accurate analysis within the capabilities of existing computers. By representing each refueling region with seven nodes (one per fuel element column) a more accurate (and expensive) analysis may also be performed without exceeding present computer capacity. RECA also provides a coupled coolant flow and temperature analysis capability for the HTGR core with CACS. RECA is the code that HEXEREI2 was designed to duplicate in analysis capability. However, HEXEREI2 is much less machine dependent than RECA, and the developmental work of HEXEREI2 has attempted to improve on the efficiency of the mathematical techniques available in RECA and the accuracy in some areas of RECA's modeling (i.e., the CACS heat exchanger).

2. GENERAL DESCRIPTION OF HEXEREI2

HEXEREI2 was developed from CCCM5 (a version of HEATING5) and retains the basic structure and some of the ancillary subroutines of CCCM5. Since HEXEREI2's mesh definition differs in basic philosophy from CCCM5, only the subroutines that were independent of the mesh definition were retained in HEXEREI2. Because of the philosophy difference and the need for a multichannel flow model, most of the major subroutines were completely rewritten. Additional subroutines were developed to enable modeling the coolant flow and CACS loop. The remainder of the code retained from CCCM5 had to be examined in detail and partially rewritten to assure compatibility with HEXEREI2's mesh definition and coolant flow model.

2.1. Capabilities of the HEXEREI2 Code

HEXEREI2 can solve three-dimensional conductive heat transfer problems with boundary conditions that include fixed temperature, forced convection, radiation, natural convection (or other nonlinear boundary conditions), or specified fluxes. All of the parameters related to the boundary conditions can be time-dependent, temperature-dependent, or both. The user may also write his own subroutines to supply these boundary conditions and, if so, can model position-dependent boundary conditions. This ability is referred to as user-supplied subroutines and provides great flexibility to the user in modeling the heat generation, boundary conditions, material properties, and initial temperatures. The use of user-supplied subroutines will be discussed further in section 3.4 and in Appendix C.

The nodal mesh that HEXEREI2 can use is restricted to a regular hexagonal mesh in the axial plane or a combination of regular hexagonal and a polar type mesh (see section 2.2). This configuration of nodes in the axial plane is repeated in all other axial planes. Thus, each node can be connected to up to six nodes in the same axial plane and one each in the adjacent axial planes. This type of mesh should be more accurate in the solution of Laplace's equation for the axial plane than a Cartesian mesh of similar grid size.

The heat generation rate can be input as a constant, or with time, temperature, and/or spatial dependence. The user-supplied subroutine technique can be used for general problems, but for HTGR-type analysis the Region Power Peaking Factor (RPPF) can also be input. Thus, the heat generated at a node can be modified by a position-, time-, and/or temperature-dependent function and further modified by a function of the HTGR refueling region in which the node lies.

HEXEREI2 provides the capability to analyze axially flowing coolant in parallel channels. The number of channels is less than or equal to the nodes in one axial plane (or numbered nodal columns). Thus, each solid node can be connected to one coolant node and up to eight other solid nodes. Each coolant node is connected to one solid node and the coolant nodes directly above and below it in the same coolant channel. The flow in each coolant channel is determined by HEXEREI2 from the relative flow retarding effects included in the channel's pressure change calculations (section 2.3.8). The coolant temperatures are determined from the flow rate and the coolant inlet and solid temperatures (section 2.3.7).

The solid node temperatures are determined from the heat generation within the node, the temperature of the surrounding solid nodes, the temperature of the one connected coolant node, and the heat capacity of the node. The equations are solved using a steady-state technique omitting the heat capacity, an explicit transient technique, or a spectrum of implicit transient techniques.

HEXEREI2 also can analyze the performance of a CACS coupled with the HTGR core thermal-hydraulic analysis. If it is desired and sufficient detail of the whole system is available, the CACS performance can supply the core inlet temperature, pressure, and flow rates so that a completely self-sufficient 'closed loop' transient analysis could be performed. Only coolant composition with respect to time would need be supplied to the present HEXEREI2 code to perform most of the accident transients that RECA has performed.

2.2. Nodal Mesh Generation

The refueling regions in the HTGRs are arranged such that each refueling region (area outlined in Fig. 2.2.1) has direct contact with the six neighboring regions. Since the minimum number of axial columns of nodes (nodal columns) that can accurately model the core is one per refueling region, HEXEREI2 has the capability of automatically generating a three-dimensional nodal mesh consisting of nodal columns arranged in a manner similar to HTGR refueling regions. This manner of nodal spacing in the axial plane (nodal plane) is called regular hexagonal mesh (also regular triangular). Richtmyer and Morton [Ref. 8] present an error estimate of the solution for heat conduction in homogeneous material using a two-dimensional regular hexagonal mesh with central finite differences. In their analysis, the regular hexagonal mesh had a smaller truncation error than a regular Cartesian mesh of the same size.

HTGR's also have irregular side reflector elements that surround the active core and produce a right circular cylinder when assembled. To accurately model this region, the nodal column placement must be modified from the regular hexagonal mesh. Since the rest of the reactor from the side reflectors outward is axisymmetric, a cylindrical mesh is used to model all but the active core and the interface between the regular hexagonal (active core) and the cylindrical meshes.

HEXEREI2's automatic mesh generation provides the user with the control of the hexagonal pitch (distance in the axial plane between nodes), the equivalent radius of the hexagonal mesh zone, the radius of axisymmetric modeling regions, and the axial spacing of the axial nodal planes. HEXEREI2 will present all information necessary to complete the analysis for the regular hexagonal mesh, the interface elements, and the cylindrical mesh. Figure 2.2.2 shows the resultant cell description of one axial plane for the active core and side reflector. Figure 2.2.3 shows one of the regular hexagonal cells with a sample connective element to an adjacent node. Figure 2.2.4 shows two sample interface cells with sample connective elements. The interface cells were developed to minimize irregular cells, provide most cell boundaries perpendicular to the heat flow paths, and allow heat flow from an interface cell to only one cylindrical mesh cell (nodes numbered 3 in Fig. 2.2.4). HEXEREI2 also is capable of reducing the number of nodes required to complete a model by consolidation of the cylindrical cells into larger cells. Figure 2.2.5 is a sample of such a possible grid and does not necessarily represent an optimum grid.

HEXEREI2 uses a node column numbering technique for one axial plane that is similar to GAC's refueling region numbering technique. The node columns are numbered from the center out by completing the concentric hexagonal rings. At the border of the active core, the outermost hexagonal rings of refueling regions are not completely filled with active refueling regions, thus producing an active core that approximates a right circular cylinder. Only the active core regions are used in the consecutive numbering sequence by GAC. HEXEREI2 continues producing full concentric hexagonal rings out to include the active core radius so that replaceable side reflectors with coolant flow could be modeled with the same technique as the interior nodes and to minimize the number of irregular cells. Thus, HEXEREI2 node

Fig. 2.2.1. HTGR Core Plane
(Ref. 4, Figure 3.1-2)

Fig. 2.2.2. HEXEREI Typical Cell Geometry for One Axial Plane

Fig. 2.2.3. Typical Hexagonal Cell with Sample Connector
from Node 1 to Node 2

Fig. 2.2.4. Typical Interface Cells with Sample Connectors

Fig. 2.2.5. Possible Cell Distribution for a Reduced Number of Nodes

column numbers correspond to GAC's refueling region numbers in the interior region of the active core but not necessarily outside of this region. Since all axial planes have the same nodal distribution, node numbering is continued with the next consecutive node number at the center of the next axial plane.

The position in the typical axial plane or the node column number is easily retrieved from the node number and is used in the coolant temperature and flow calculations and to determine any function that is dependent on node column number (i.e., region power peaking factor). Thus, the node column number is used to relate the model parameters to the input information on the HTGR being analyzed. The user must exercise care when inputting the data in that the region number used is HEXEREI2's node column number and not GAC's refueling region number. Since all of the regular hexagonal and partial hexagonal interface cells (Figs. 2.2.3 and 2.2.4) have nodes that fall on a regular hexagonal mesh, it should not be too difficult to relate HEXEREI2's nodes with GAC's refueling regions (see Fig. 2.2.6).

HEXEREI2 is capable of generating a mesh with any node column hexagonal pitch, so it is possible to geometrically model an HTGR with a finer grid than one nodal column per refueling region. Seven, or a multiple of seven, nodal columns per refueling region would be desirable to most accurately model the HTGRs within the constraints of the computing system utilized. At present, each nodal column is assumed to be an independent flow channel from the top to the bottom of the model, thus coding modifications would be necessary to provide for several flow channels sharing entrance and exits. GAC's designs currently provide each refueling region with one orifice at the region's inlet from the upper plenum and the coolant channels for one refueling region are merged into a common exit. In addition, the user must relate with care GAC's refueling region data to HEXEREI2's computational mesh to effect the more accurate analysis.

HEXEREI2 will automatically produce all the thermal and fluid dynamic model parameters necessary for the chosen analyses with the mesh generated by the above procedure.

2.3. Mathematical Description

2.3.1. Statement of the Problem

The HEXEREI2 Program solves the steady state or transient heat conduction problem in three dimensions for cylindrical coordinates using hexagonal and cylindrical mesh. First, the physical problem is approximated by a system of nodes, described in the previous section, each associated with a small volume or cell. A typical internal node and cell are depicted in Fig. 2.3.1.1. For the three-dimensional problem, heat may flow from an internal node to each of its eight neighboring nodes plus its related coolant node. The system of equations describing the temperature distribution is derived by performing a heat balance about each node.

The finite difference heat balance equation for solid node o in Fig. 2.3.1.1 is

$$C_o \frac{T_o^{n+1} - T_o^n}{\Delta t} = P_o^n + \sum_{m=1}^8 {}_oK_m(T_m^n - T_o^n) + B_o^n(T_{c_o}^n - T_o^n) \quad (2.3.1.1)$$

where T_m^n is the temperature of the m th node adjacent to node o at time t^n , ${}_oK_m$ is the conductance between nodes o and m , C_o is the heat capacitance of the material associated with node o , $T_{c_o}^n$ is the temperature of the coolant node associated with node o at time t^n , B_o^n is the effective conductivity for convection from node o to the coolant node at time t^n , and P_o^n is the heat generation rate in the cell for node o at time t^n . Since HEXEREI2 utilizes the cell center nodal technique, the material for a node is homogeneous and the heat flow path between

Fig. 2.2.6. Comparison of GAC and HEXEREI2 Core Maps

Fig. 2.3.1.1. Typical Solid Node

adjacent nodes may be composed of two different materials positioned in series. For a normal problem, one C, one P, one B, and eight K's will be associated with each internal node at a particular time, t_n . For conduction only problems, the coolant heat transfer is simply neglected. These parameters are calculated as follows for node o:

$$C_o = C_p \rho V \quad (2.3.1.2)$$

$$P_o^n = q^n V \quad (2.3.1.3)$$

$${}_oK_m = \frac{1}{\frac{L_o}{A_m K_o} + \frac{L_m}{A_m K_m}} \quad (2.3.1.4)$$

$$B_o^n = h_o A_{co} \quad (2.3.1.5)$$

where

C_p = specific heat for node o,

ρ = density for node o,

V = volume of cell for node o,

q^n = heat generation rate per unit volume in cell at time t_n ,

L_o = distance between node o and cell boundary adjacent to node m,

L_m = distance between node m and cell boundary adjacent to node o,

k_m = thermal conductivity for node m,

k_o = thermal conductivity for node o,

A_m = cross-sectional area, normal to heat flow path, between nodes o and m,

h_o = convective heat transfer coefficient for node o,

A_{co} = surface area for convective heat transfer for node o.

Since nodes lying on a surface of the model or nodes in the cylindrical mesh zone will not necessarily have eight neighbors, the general heat balance equations for node i having M_i neighbors can be written as

$$C_i \frac{T_i^{n+1} - T_i^n}{\Delta t} = P_i^n + \sum_{m=1}^{M_i} {}_iK_{\alpha_m} (T_{\alpha_m}^n - T_i^n) + B_i^n (T_{C_i}^n - T_i^n) \quad (2.3.1.6)$$

where α_m is the mth neighbor of the ith node. By choosing the mesh spacing small enough, the solution to the system of equations yields a practical approximation to the appropriate differential equation.

2.3.2. Steady-State Heat Conduction

For a steady-state heat conduction problem, the heat balance equation reduces to

$$P_i + \sum_{m=1}^{M_i} K_{\alpha_m} (T_{\alpha_m} - T_i) + B_i (T_{c_i} - T_i) = 0 \quad (2.3.2.1)$$

since the left-hand side of Eq. (2.3.1.6) is zero.

If there are I nodal points, then since Eq. (2.3.2.1) will be applied at each node, there will be a system of I equations with I unknowns. The iterative technique which is used by HEXEREI2 to solve the system of equations is outlined below. First, solve Eq. (2.3.2.1) for T_i .

$$T_i = \frac{P_i + \sum_{m=1}^{M_i} K_{\alpha_m} T_{\alpha_m} + B_i T_{c_i}}{\sum_{m=1}^{M_i} K_{\alpha_m} + B_i} \quad (2.3.2.2)$$

Since the values of T_{α_m} and T_{c_i} are unknown, the temperature at node i cannot be calculated directly from Eq. (2.3.2.2). However, an iterative procedure based on Eq. (2.3.2.2) can be used to estimate the steady-state temperature distribution. If an estimate to the temperature distribution exists, then Eq. (2.3.2.2) can be applied at each node and, hopefully, a better estimate to the temperature distribution will be obtained. Then, this new estimate can be used in Eq. (2.3.2.2) to produce an even better estimate. This iterative process can be written as

$$T_i^{(n+1)} = \frac{P_i + \sum_{m=1}^{M_i} K_{\alpha_m} T_{\alpha_m}^{(n)} + B_i^{(n)} T_{c_i}^{(n)}}{\sum_{m=1}^{M_i} K_{\alpha_m} + B_i^{(n)}} \quad (2.3.2.3)$$

where the superscript (n) implies the n th iterate. The coolant temperatures are solved by the technique described in section 2.3.6 for each iteration. Instead of using the results of Eq. (2.3.2.3) as the $(n+1)$ st iterate, assume that it only yields an estimate and denote it as $\hat{T}_i^{(n+1)}$. Then define the $(n+1)$ st iterate to be

$$T_i^{(n+1)} = T_i^{(n)} + \beta [\hat{T}_i^{(n+1)} - T_i^{(n)}] \quad (2.3.2.4)$$

where the relaxation factor, β , is limited to $0 < \beta < 2$. Notice that when $\beta > 1$, the new iterate is changed more than Eq. (2.3.2.3) specifies and, thus, the iterate is overrelaxed. Likewise, when $\beta < 1$, the iterate is underrelaxed. If $\hat{T}_i^{(n+1)}$ is replaced by Eq. (2.3.2.3), then Eq. (2.3.2.4) can be written as

$$T_i^{(n+1)} = (1-\beta)T_i^{(n)} + \beta \left[\frac{P_i - \sum_{m=1}^{M_i} K_{\alpha_m} T_{\alpha_m}^{(n)} + B_i^{(n)} T_{C_i}^{(n)}}{\sum_{m=1}^{M_i} K_{\alpha_m} + B_i^{(n)}} \right] \quad (2.3.2.5)$$

To increase the rate of convergence for the iterative procedure, the most recent values of the temperatures of the nodes are used in Eq. (2.3.2.5). This algorithm can be written as

$$T_i^{(n+1)} = (1-\beta)T_i^{(n)} + \beta \left[\frac{P_i + \sum_{m=1}^{L_i} K_{\alpha_m} T_{\alpha_m}^{(n+1)} + \sum_{m=L_i+1}^{M_i} K_{\alpha_m} T_{\alpha_m}^{(n)} + B_i^{(n)} T_{C_i}^{(n)}}{\sum_{m=1}^{M_i} K_{\alpha_m} + B_i^{(n)}} \right] \quad (2.3.2.6)$$

where L_i is defined so that $\alpha_{L_i} < i < \alpha_{L_i+1}$. Varga [Ref. 9] refers to this method as the point successive overrelaxation iterative method. To increase the rate of convergence, an exponential approximation for Eq. (2.3.2.6) is made based on the temperature change from one iteration to the next. The algorithm based on this approximation is used instead of Eq. (2.3.2.6) to calculate the new temperatures for nodes having relative temperature changes exceeding 10^{-1} . However, the algorithm is designed to bound the temperature change so that the new temperature cannot be more than two times the old temperature. This prohibits the technique from diverging due to a bad estimate of the initial temperature distribution. The exponential approximation reduces to Eq. (2.3.2.6) for small temperature changes. Successive iterations are carried out by HEXEREI2 until

$$\left| \frac{T_i^{(n)} - T_i^{(n+1)}}{T_i^{(n+1)}} \right| \leq \epsilon \quad , \quad (2.3.2.7)$$

where ϵ is the specified convergence criterion.

Another extrapolation procedure which is commonly used to increase the rate of convergence in an iterative solution to a system of equations is the "Aitken δ^2 extrapolation procedure." Briefly, if $T^{(n-1)}$, $T^{(n)}$, and $T^{(n+1)}$ are the temperatures at a certain point at the $n-1$ st, n th and $n+1$ st iterations, respectively, and if

$$|T^{(n)} - T^{(n-1)}| > |T^{(n+1)} - T^{(n)}| \quad (2.3.2.8)$$

and

$$[T^{(n+1)} - T^{(n)}] [T^{(n)} - T^{(n-1)}] > 0 \quad , \quad (2.3.2.9)$$

then, a better estimate of the temperature is

$$T_{new} = T^{(n+1)} + \frac{[T^{(n+1)} - T^{(n)}]}{[T^{(n)} - T^{(n-1)}] - [T^{(n+1)} - T^{(n)}]} \quad (2.3.2.10)$$

Actually HEXEREI2 uses a modification of the Aitken's δ method by calculating an extrapolation factor, B and approximating Eq (2.3.2.10) at each node with

$$T_i^{(n+1)} = T_i^{(n+1)} - B[T_i^{(n+1)} - T_i^{(n)}] \quad , \quad (2.3.2.11)$$

where $T_i^{(n-1)}$ represents the $n+1$ st iterate at node i before extrapolation

A HEXEREI2 extrapolation cycle is defined as follows. The code completes 20 iterations and checks to see if the maximum of the absolute values of the relative temperature changes over an iteration has decreased monotonically over the last ten iterations. If not, the cycle starts over. If so, the code will extrapolate only if the relative change in extrapolation factors over two consecutive iterations is less than 5% and the maximum of the absolute values of the relative temperature changes decreases monotonically over the same two iterations. The extrapolation factor, B, which is the same for each node, is based on two maximum relative temperature changes: 1) between the $n-1$ st and the n th iterations and, 2) between the n th and $n+1$ st iterations.

The value of β in Eq (2.3.2.6) which will produce the optimum convergence rate for all points is difficult to obtain analytically for simple geometries and is practically impossible to obtain for complex geometries. If an input value is not supplied for β , then HEXEREI2 uses the default value of 1.9. If the rate of convergence appears to be slow, then HEXEREI2 reduces β by 0.1. The code determines whether or not the rate of convergence is slow in the following manner. It was noted above that during an extrapolation cycle, the relative temperature change over an iteration is monitored over ten consecutive iterations. If the relative temperature changes do not decrease monotonically over these ten iterations, then the current relative temperature change is compared with the one arising ten iterations earlier. If the current relative temperature change is greater than two-thirds of the old one, then the SOR technique is converging slowly. This process may be repeated until $\beta = 1.0$. However, the code will not increase β .

2.3.3 Transient Heat Conduction

HEXEREI2 is designed to solve a transient problem by any one of several numerical schemes. The first is the Classical Explicit Procedure (CEP) which involves the first forward difference with respect to time and is thus stable only when the time step is smaller than the stability criterion. Levy's modification to the CEP is the second scheme, and it requires the temperature distribution at two times to calculate the temperatures at the new time level. The technique is stable for a time step of any size. The third transient procedure, which is written quite generally, actually contains several implicit techniques which are stable for a time step of any size. One can use the Crank-Nicolson heat balance equations, the Classical Implicit Procedure (CIP) or backwards Euler heat balance equations or a linear combination of the two. The resulting system of equations is solved by point successive overrelaxation iteration. Techniques have been included in the code to approximate the optimum acceleration parameter for problems involving constant thermal parameters as well as those whose effective thermal conductances and capacitances vary with time or temperature.

The stability criterion for the CEP is a function of a temperature-dependent heat generation rate or heat flux. This fact is not accounted for in HEXERE12. Also Levy's modification to the CEP is based on a constant heat generation rate or heat flux with respect to temperature. If one attempts to use one of the explicit transient algorithms along with a temperature-dependent source or heat flux, then the code will write out a warning message indicating that the time step allowed by HEXERE12 may not yield a stable solution. Levy's modification has not been reevaluated for problems involving coolant and its use is not recommended for these problems.

Equation (2.3.1.6) is the basic heat balance equation for transient problems. However, the right-hand side is modified for all but the CEP scheme which was discussed above.

2.3.3.1 Classical Explicit Procedure

For a transient heat conduction problem, the heat balance equation, Eq. (2.3.1.6), can be solved for T_i^{n+1} to give

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{C_i} \left[P_i^n + \sum_{m=1}^{M_i} K_{\alpha_m} (T_{\alpha_m}^n - T_i^n) \right] + B_i^n (T_{c_i}^n - T_i^n) \quad (2.3.3.1)$$

Since Eq. (2.3.3.1) expresses the temperature of the i th node at the $n+1$ st time level in terms of temperatures at the n th time level, it is an explicit technique, and the algorithm is known as the Classical Explicit Procedure (CEP) or the Forward Difference Equation. HEXERE12 uses Eq. (2.3.3.1) to solve transient heat conduction problems. The numerical solution obtained by using this technique is stable, provided the time step satisfies the following equation [Ref. 10]

$$\Delta t \leq \left(\frac{C_i}{\sum_{m=1}^{M_i} K_{\alpha_m}} \right)_{\text{minimum for all nodes}} \quad (2.3.3.2)$$

2.3.3.2 Levy's Modification to the Classical Explicit Procedure

The limitation of the size of the time step as indicated in Eq. (2.3.3.2) means, in many practical problems, a very high ratio of computer time to actual time. In some cases computation costs become so high that the use of the algorithm defined by Eq. (2.3.3.1) becomes impractical. Levy [Ref. 11] proposed a modified explicit method which is stable for any time step desired. This method has been incorporated as an option in HEXERE12 for conduction problems, and the method has been used extensively in HEATING5. The method is not recommended for problems with coolant flow until analysis is concluded on the method's appropriateness for such problems. The basic equation used is

$$T_i^{n+1} = T_i^n + \frac{1}{1+Z_i} \left\{ \frac{\Delta t}{C_i} \left[\sum_{m=1}^{M_i} K_{\alpha_m} (T_{\alpha_m}^n - T_i^n) + P_i^n \right] + Z_i (T_i^n - T_i^{n-1}) \right\} \quad (2.3.3.3)$$

where

Z_i = a factor for node i which will insure a stable solution for any time step Δt

If from Eq. (2.3.3.2), $(\Delta t_{max})_i$ is the maximum time step allowed at node i for a stable solution in the regular explicit method, then the factor Z_i in Eq. (2.3.3.3) is defined as

$$Z_i = \begin{cases} 0, & \text{if } \frac{\Delta t}{(\Delta t_{max})_i} \leq 1 \\ 0.5 \left[\frac{\Delta t}{(\Delta t_{max})_i} - 1 \right], & \text{if } \frac{\Delta t}{(\Delta t_{max})_i} > 1 \end{cases} \quad (2.3.3.4)$$

Levy says that the accuracy is good if Z_i is zero for somewhat over half of the nodes. Of course, one must experiment with the size of the time step in order to obtain an acceptable solution.

2.3.3.3 Implicit Transient Numerical Procedure

a. Heat Balance Equation

If the right-hand side of Eq. (2.3.1.6) is evaluated at t_{n+1} instead of t_n , then the scheme is known as the Classical Implicit Procedure (CIP) or the backwards Euler procedure. If the right-hand side of Eq. (2.3.1.6) is evaluated at $t_{n+1/2}$, then the algorithm is known as the Crank Nicolson (CN) procedure. A general algorithm which includes both the CN technique and the CIP is

$$\begin{aligned} C_i^{n+\Theta} \frac{T_i^{n+1} - T_i^n}{\Delta t} = & P_i^{n+\Theta} + \Theta \left[\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\Theta} (T_{\alpha_m}^{n+1} - T_i^{n+1}) \right] \\ & + (1-\Theta) \left[\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\Theta} (T_{\alpha_m}^n - T_i^n) \right] + B_i^{n+1} [T_{c_i}^{n+1} - T_i^{n+1}] \end{aligned} \quad (2.3.3.5)$$

where $0 \leq \Theta \leq 1$ and where the superscript $n+\Theta$ implies that the parameter is evaluated at time $t_{n+\Theta}$. If $\Theta = 0.5$, then Eq. (2.3.3.5) becomes the CN technique and if $\Theta = 1.0$, the algorithm is the CIP. When Θ is less than 0.5, the technique is no longer stable for any time step. Notice that Eq. (2.3.3.5) reverts to Eq. (2.3.1.6) when $\Theta = 0$. This algorithm has been incorporated into HEXEREI2 for $0.5 \leq \Theta \leq 1.0$.

b. Numerical Technique

If there are I nodes in the problem and if the heat balance equation, Eq. (2.3.3.5), is written for each node, then there will be I equations and I unknowns, and the resulting system of equations can be solved iteratively. The procedure that is used in HEXEREI2 is outlined below. If Eq. (2.3.3.5) is rewritten so that the temperatures at t_{n+1} are on the left-hand side, then the equation becomes

$$-\Theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\Theta} T_{\alpha_m}^{n+1} \right) - B_i^{n+1} T_{c_i}^{n+1} + \left[\frac{C_i^{n+\Theta}}{\Delta t} + \Theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\Theta} \right) + B_i^{n+1} \right] T_i^{n+1} = H_i \quad (2.3.3.6)$$

where

$$H_i = \frac{C_i^{n+\theta}}{\Delta t} T_i^n + P_i^{n+\theta} + (1-\theta) \left[\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\theta} (T_{\alpha_m}^n - T_i^n) \right] \quad (2.3.3.7)$$

If we let

$$D_i = \frac{C_i^{n+\theta}}{\Delta t} + \theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\theta} \right) + B_i^{n+1} \quad (2.3.3.8)$$

and if we delete the superscript, $n+1$, on the temperature, T , then Eq. (2.3.3.6) can be rewritten as

$$-\theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\theta} T_{\alpha_m} \right) - B_i^{n+1} T_{c_i}^{n+1} + D_i T_i = H_i \quad (2.3.3.9)$$

where it is now understood that T_i represents the temperature of node i at the new time level. If Eq. (2.3.3.9) is solved for T_i , then

$$T_i = \frac{\theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\theta} T_{\alpha_m} \right) + B_i^{n+1} T_{c_i}^{n+1} + H_i}{D_i} \quad (2.3.3.10)$$

Since the values of T_{α_m} are unknown, one cannot directly solve for the temperature at node i . However, if an estimate of the temperature distribution at the new time level exists, then Eq. (2.3.3.10) can be solved at each node and, hopefully, one will have a better estimate for the temperature distribution. The procedure can be repeated using this better estimate. This process can be continued until the estimates have converged to the approximation to the temperature distribution at the new time level. This algorithm can be written as

$$T_i^{(n+1)} = \frac{\theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\theta} T_{\alpha_m}^{(n)} \right) + B_i^{(n)} T_{c_i}^{(n)} + H_i}{D_i} \quad (2.3.3.11)$$

where the superscript (n) on T_i refers to the n th iterate of the temperature at node i at the new time level.

Instead of using Eq. (2.3.3.11) in the iterative process, the technique can be refined further. First, consider Eq. (2.3.3.11) as only an approximation to the (n+1)st iterate or

$$\hat{T}_i^{(n+1)} = \frac{\Theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\Theta} T_{\alpha_m}^{(n)} \right) + B_i^{(n)} T_{c_i}^{(n)} + H_i}{D_i} \quad (2.3.3.12)$$

and then define the (n+1)st iterate as

$$T_i^{(n+1)} = T_i^{(n)} + \omega [\hat{T}_i^{(n+1)} - T_i^{(n)}] \quad (2.3.3.13)$$

or

$$T_i^{(n+1)} = (1-\omega) T_i^{(n)} + \omega \hat{T}_i^{(n+1)} \quad (2.3.3.14)$$

where $0 < \omega < 2$. Thus, the change in temperature based on the value at the last iteration is more ($\omega > 1$) or less ($\omega < 1$) than the calculated value. Usually, the iterative procedure converges faster than $\omega \neq 1$, and thus, ω is commonly referred to as the acceleration parameter. Combining Eqs. (2.3.3.12) and (2.3.3.14), one obtains

$$T_i^{(n+1)} = (1-\omega) T_i^{(n)} + \omega \left[\frac{\Theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\Theta} T_{\alpha_m}^{(n)} \right) + B_i^{(n)} T_{c_i}^{(n)} + H_i}{D_i} \right] \quad (2.3.3.15)$$

If we always use the most recent iterates on the right-hand side of Eq. (2.3.3.15), then the algorithm reduces to

$$T_i^{(n+1)} = (1-\omega) T_i^{(n)} + \omega \left\{ \frac{\Theta \left[\left(\sum_{m=1}^{L_i} K_{\alpha_m}^{n+\Theta} T_{\alpha_m}^{(n+1)} \right) + \left(\sum_{m=L_i+1}^{M_i} K_{\alpha_m}^{n+\Theta} T_{\alpha_m}^{(n)} \right) \right] + B_i^{(n)} T_{c_i}^{(n)} + H_i}{D_i} \right\} \quad (2.3.3.16)$$

where $\alpha_{L_i} < i < \alpha_{L_i+1}$. It was pointed out in the previous section that this method is referred to as the point successive overrelaxation iterative method. HEXEREI2 applies Eq. (2.3.3.16) to each node until the convergence criteria have been met.

The convergence criteria are derived as follows. When the nth iteration has been completed, substitute the nth iterates into Eq. (2.3.3.9) and denote the residual as

$$R_i^{(n)} = H_i - B_i T_{c_i} + \Theta \left(\sum_{m=1}^{M_i} K_{\alpha_m}^{n+\Theta} T_{\alpha_m}^{(n)} \right) - D_i T_i^{(n)} \quad (2.3.3.17)$$

Now, consider the L_1 norm of the residual or

$$\|R^{(n)}\|_1 = \sum_{i=1}^I |R_i^{(n)}| \quad . \quad (2.3.3.18)$$

HEXEREI2 uses two convergence criteria based on the L_1 norm of the residual. They are

$$\|R^{(n)}\|_1 \leq \epsilon_1 \quad (2.3.3.19)$$

and

$$\frac{\|R^{(n)}\|_1}{\|R^{(1)}\|_1} \leq \epsilon_2 \quad . \quad (2.3.3.20)$$

The iterative procedure is started in the following manner. For the first time step, the starting estimate is equal to the initial temperature distribution. Thereafter, the starting estimate at t_{n+1} is determined by

$$\hat{T}_i^{n+1} = T_i^n + (T_i^n - T_i^{n-1})(\Delta t_{n+1}/\Delta t_n) \quad . \quad (2.3.3.21)$$

c. Temperature-Dependent Properties

For problems involving temperature-dependent thermal properties, the iterative procedure is as follows. Initially, the thermal parameters are evaluated at the initial temperatures. Then, the point successive overrelaxation iterative method as described above is applied to obtain the temperature distribution at the new time level. However, since none of the thermal parameters are updated during this procedure, the converged temperatures are only an estimate to the temperature distribution. Thus, the thermal properties are reevaluated and the entire procedure is repeated until the technique converges to the temperature distribution at the new time level. This process contains two levels of iteration. The inner loop is the basic iterative process in the point successive overrelaxation iterative method while the outer loop iterates on the thermal parameters. Let $\hat{T}_i^{n,m}$ denote the temperature of node i after the m th iteration on the outer loop at time t_n . Upon the completion of the m th outer loop, the temperature at which the thermal parameters are evaluated is calculated as

$$\hat{T}_i^{n+\Theta} = (1 - \theta)T_i^n + \theta \hat{T}_i^{n+1,m} \quad . \quad (2.3.3.22)$$

The temperature distribution has converged at time t_n when the L_1 norm of the temperature difference over successive iterations is less than the prescribed value or

$$\|\Delta T^{m,m}\|_1 \leq \epsilon, \quad (2.3.3.23)$$

where

$$\|\Delta T^{n,m}\|_1 = \sum_{i=1}^I |T_i^{n,m} - T_i^{n,m-1}| \quad (2.3.3.24)$$

d. Estimation of the Optimum Acceleration Parameter

The rate of convergence of the point successive overrelaxation iterative method is strongly dependent on the value of the acceleration parameter ω . The optimum value of the parameter, denoted as ω_o , is usually not known prior to the solution to a problem, and it is a function of time for problems whose effective thermal conductances and capacitances vary with time or temperature. Several techniques have been developed to estimate ω_o for transient problems with constant thermal properties. The method developed by Carré [Ref. 12] has been incorporated into HEXEREI2. Briefly, this method consists of estimating ω_o based on the behavior of a norm of the residual vector during the iterative procedure. The estimates are computed as a function of the iteration number until the process converges to a best estimate to the optimum value. Thereafter, the code uses this converged value as the acceleration parameter.

It was observed that this process was not satisfactory for problems involving temperature- and time-dependent conductances and capacitances, so an empirical process was developed and added to HEXEREI2 to estimate ω_o . For the initial time step, ω is equal to unity. The code will attempt to update ω every N_ω time steps relative to the last time ω was changed. The criteria which are applied to determine whether or not the current value of ω is a good estimate to ω_o are based on the number of iterations required for the inner iterative loop to converge on the first pass through the outer iterative loop at some time step. When the code determines that an attempt to update ω should be made after completion of a particular time step, then the number of iterations for this time step is compared to the number for the time step immediately following the last modification to ω . If the change in the number of iterations is equal to or exceeds the criterion I_ω (an input value), then ω is increased according to

$$\omega^{n+1} = \omega^n + 0.1 (2.0 - \omega^n) \quad (2.3.3.25)$$

where the superscript n refers to the value of ω at time t_n . On each subsequent time step a new estimate is made for ω_o using an algorithm similar to Eq. (2.3.3.25). However, ω may be either increased or decreased according to Table 2.1. When the change in the number of iterations for two consecutive time steps is less than the criterion J_ω (an input value), the code assumes that it has a good estimate for ω_o , and it uses this value for the subsequent time steps until it is time to attempt another ω update. At this time, the entire procedure is repeated.

Table 2.1. Logic to Determine Whether ω Should be Increased or Decreased

Last update resulted in ω being	Change in number of iterations compared to previous time step	Increase	Decrease
		Increase	Decrease
Increased		Decrease	Increase
Decreased		Increase	Decrease

2.3.3.4 Variable Time Step Size

The time step size can be varied in a number of ways during the implicit transient calculations. First it can be controlled explicitly by specifying it as a constant size during the entire calculation or during prescribed time intervals throughout the transient. The time step size can also be varied explicitly by multiplying it by some prescribed factor after each time step subject to maximum and minimum values which may or may not be specified.

Finally, it can be varied implicitly in two ways. One can specify the maximum temperature change and the maximum percent of relative change in temperature allowed at a node over a time step. The time step size is either decreased if one of the calculated maximum values exceeds its respective criterion or is increased if both of the calculated maximum values are less than their respective criteria. The size is increased or decreased in the following manner. If one of the calculated values exceeds its respective criterion, then

$$f_{\Delta t} = \min \left(0.95 * \frac{\text{criterion}}{\text{calculated value}}, \text{TSFACT} \right)$$

where TSFACT is an input value representing the factor which is multiplied by the old time step size to obtain the new time step size. The factor 0.95 is to insure the decrease is large enough since the maximum temperature change is not really linear with changing time step size. The new time step size is then determined as

$$\Delta t_{\text{new}} = \max(\Delta t_{\text{min}}, f_{\Delta t} * \Delta t_{\text{old}})$$

where Δt_{min} is an input value representing the smallest time step size allowed and Δt_{old} is the current time step size. If one of the calculated values is less than its respective criterion, then

$$f_{\Delta t} = \min \left(0.95 * \frac{\text{criterion}}{\text{calculated value}}, 1.0 \right)$$

where the factor 0.95 is to insure the increase is not too large since maximum temperature change is not really a linear function of time step size. Then

$$f_{\Delta t} = \min (f_{\Delta t}, \text{TSFACT}, 2.0)$$

where TSFACT was described above. The new time step size is then calculated as

$$\Delta t_{\text{new}} = \min (\Delta t_{\text{max}}, f_{\Delta t} * \Delta t_{\text{old}})$$

where Δt_{max} is an input value representing the largest time step size allowed. If both options are specified, the code uses the smaller of the two new time steps.

If the new time step size is smaller than the old one, the code rejects the temperature distribution it has just calculated and returns to the old time level. It then calculates a new temperature distribution at the new time level using the smaller time step size. If the code reduces the time step size NREP (currently 10) times, it writes out a warning message, reduces the time step size to the minimum value, recalculates the temperature distribution using the new time step size and moves ahead to the next time level.

If the new time step size is greater than or equal to the old one, the code accepts the temperature distribution it has just calculated and moves on to the next time level. If a time step size has been reduced, the code will not allow it to be increased again until NREDUC (currently equal to 5) time steps have lapsed.

When the time step size is varied implicitly, the time step size and the associated criteria can be greatly affected by discontinuities in the boundary conditions, time-dependent functions, temperature-dependent functions, and especially the initial conditions. The user must be selective in choosing input parameters for these cases. One approach would be to control the time step size explicitly for short periods of time following these discontinuities. One may also need to start out with small time step size initially to smooth out the data.

As the transient calculations approach a printout time, the time step size is automatically reduced to allow the temperature distribution to be printed out at the exact specified time. The old time step size is saved so that calculations can resume using the old time step after the printout.

2.3.4. Temperature-Dependent Thermal Properties

HEXEREI2 allows the thermal conductivity, k , the specific heat, C_p , and the density, ρ , to vary with temperature. The code determines the conductivity of the material between two nodes, i and j , by evaluating the temperature-dependent conductivity at an average of the temperatures of the two nodes. This temperature is calculated as

$$\bar{T}^{(n)} = \frac{T_i^{(n-1)} + T_j^{(n-1)}}{2} \quad (2.3.4.1)$$

after completion of the nth iteration for steady-state problems, as

$$\bar{T}^n = \frac{T_i^{n-1} + T_j^{n-1}}{2} \quad (2.3.4.2)$$

after completion of the nth time step for transient problems involving one of the explicit techniques, and as

$$\bar{T}^{n+\Theta} = \frac{\hat{T}_i^{n+\Theta} + \hat{T}_j^{n+\Theta}}{2} \quad (2.3.4.3)$$

during the calculation of the temperature distribution at time t_{n+1} for transient problems involving the implicit procedure. The temperatures denoted as $\hat{T}_i^{n+\Theta}$ in Eq. (2.3.4.3) are evaluated according to Eq. (2.3.3.22). For transient problems, the specific heat and density are determined for node i by evaluating the respective temperature-dependent function at T_i^n after completing the nth time step using one of the explicit techniques and at $\hat{T}_i^{n+\Theta}$ as determined by Eq. (2.3.3.22) during the calculation of the temperature distribution at time t_{n+1} using the implicit procedure.

2.3.5. Boundary Conditions

HEXEREI2 possesses a variety of boundary conditions to enable the user to model his physical problem as accurately as possible. In general, a boundary condition is applied along a surface of a region and heat is transferred from a surface node to a boundary node. Surface nodes are actually internal nodes which are not completely surrounded by materials. Boundary nodes are dummy nodes and only their temperatures enter into the calculations. The boundary conditions which can be applied over the surface of a region in the current version of HEATING5 are listed below.

- a. The temperature on the surface of a region can be specified as a constant or a function of time.
- b. The heat flux across the surface of a region can be specified directly as a constant or a function of time and/or surface temperature.
- c. The heat flux across the surface of a region can be specified indirectly by defining the heat transfer mechanism to be forced convection, radiation and/or natural convection.

The numerical techniques used in calculating the temperatures of surface nodes associated with a boundary condition are discussed below. The temperatures of nodes on surfaces whose temperatures are specified are not calculated from Eq. (2.3.1.6). Instead, the surface node temperature is set equal to the specified value. When a heat flux is specified across a surface, then for each node on that surface, the specified heat flux is multiplied by its surface area normal to the heat flow path associated with the boundary condition, and the result is added to the heat generation term, P_i^n , in Eq. (2.3.1.6). If the heat flux is temperature-dependent, then it is evaluated at the average temperature of the related surface node and boundary node. Simulation is not required for insulated boundaries. Heat is simply not allowed to cross the surface.

The boundary conditions are classed as either surface-to-boundary or isothermal. Boundary conditions of the surface-to-boundary type are used to define heat transfer between a surface node and a boundary node. The temperature of the boundary node is specified and can be a function of time.

For surface-to-boundary conditions, the leakage term in Eq. (2.3.1.6) is calculated as follows:

$$[{}_iK_{\alpha_m}(T_{\alpha_m}^n - T_i^n)] = {}_iK_b(T_b^n - T_i^n) \quad (2.3.5.1)$$

where ${}_iK_b$ is the effective conductance from surface node i to boundary node b , and T_b^n is the temperature of boundary node b at time t_n . The effective conductance is calculated as

$${}_iK_b = hA \quad (2.3.5.2)$$

where h is the effective heat transfer coefficient, and A is the surface area, normal to the heat flow path, of node i associated with the boundary condition.

The effective heat transfer coefficient is defined as

$$h = h_c + h_r [(T_i^n)^2 + (T_b^n)^2] [T_i^n + T_b^n] + h_n |T_i^n - T_b^n|^{h_e} \quad (2.3.5.3)$$

where

- h_c is the heat transfer coefficient for forced convection,
- h_r is a coefficient (product of the gray body shape factor and Stefan-Boltzman constant) for radiative heat transfer,
- h_n and h_e are, respectively, the coefficient and exponent for the term simulating the effects of natural convective heat transfer or some other heat transfer process such as boiling.

The parameters h_c , h_r , h_n , and h_e must be specified by the user and can be time- and/or temperature-dependent. When h_c , h_n , and h_e are temperature-dependent, they are evaluated at the average temperature of the related surface node and boundary node, whereas h_r is evaluated at the temperature of the surface node. For surface-to-boundary boundary conditions, the boundary temperature, T_b^n , must also be supplied by the user. The temperatures are entered in either °F or °C, and the code converts them to absolute degrees when a radiative boundary condition exists.

2.3.6. Calculation of Coolant Temperatures

Figure 2.3.6.1 contains a simplified presentation of a coolant channel cell and solid node as used in the coolant temperature calculation technique. Figure 2.3.6.2.a presents a typical HTGR fuel element with coolant channels and Fig. 2.3.6.2.b presents a control rod element. For the coolant temperature calculations, HEXEREI2 models this complex assembly of parallel channels containing two different diameters with a single channel with the diameter of the larger channels which comprise the vast majority of the channels. The mass flux is assumed to

Fig. 2.3.6.1. Coolant and Solid Cells

Fig. 2.3.6.2.a Regular Fuel Elements

Fig. 2.3.6.2.b Control Rod Fuel Elements

be equal for all channels in a refueling region. Thus, the flow in the model channel is adjusted (reduced slightly) to approximate the effect of the flow down the smaller channels and to assure conservation of total flow.

In the present technique, the coolant inlet temperature is the exit temperature from the adjacent upstream coolant cell and the only resistance to heat flow from the coolant node to the solid node (which is the solid cell volume average temperature) is the coolant film resistance. Since the temperature gradient from the volume average temperature to the solid surface of the coolant channels during reactor trip accidents is small and because a detailed analysis of the fine structures of the fuel element is needed to approximate the resistance to heat flow from the volume average temperature to the surface temperature, this resistance is omitted in the present technique. If more detailed knowledge of this resistance is available for a specific model, it can be added to the technique with relatively minor coding changes.

The steady-state solution for coolant flowing through a tube with a constant wall temperature is used to find, first, the temperature of the coolant exiting the cell, and then the equivalent coolant temperature which is reported as the coolant node temperature.

Equation 2.3.6.1 represents the solution for the coolant axial temperature distribution for a coolant cell.

$$T_c(z) = T_s - (T_{c(0)} - T_s) e^{-\left(\frac{hA(z)}{G C_p A_c}\right)} \quad (2.3.6.1)$$

where

h = film heat transfer coefficient

A_c = surface area of coolant-solid interface

z = (axial distance from cell inlet)(circumference of interface)

A_c = coolant flow area

T_c = coolant temperature

T_s = solid temperature

G = mass flux of coolant

C_p = specific heat of coolant.

The coolant exit temperature is found from Eq.(2.3.6.1) by replacing Z with the cell length. The solution is graphically represented in Fig. 2.3.6.3.

The total heat transferred from solid to coolant is as shown in Eq.(2.3.6.2).

$$\begin{aligned} q &= h \int_0^{l_z} C [T_s - (T_s - (T_{c(0)} - T_s) e^{-\left(\frac{hA(z)}{G C_p A_c}\right)})] dZ \\ &= G C_p A_c (T_s - T_{c(0)}) (e^{-\left(\frac{hA}{G C_p A_c}\right)} - 1) \end{aligned} \quad (2.3.6.2)$$

Fig. 2.3.6.3. Graphical Representation of Coolant Temperature Solution

where

l_z = cell axial length

A = total surface area

c = circumference of solid-coolant interface.

The coolant node temperature (T_c) is as defined in Eq. 2.3.6.3

$$q = hA(T_s - \bar{T}_c) \quad (2.3.6.3)$$

the coolant node temperature can be found by Eq. 2.3.6.4

$$\bar{T}_c = T_s - \frac{q}{hA} \quad (2.3.6.4)$$

Figure 2.3.6.4 presents the whole length of the coolant channel and solid node model used in the present technique. The axial coordinate is shown, as used in HEXEREI2, to be positive in the normal flow direction. Thus, the top of the reactor is located at $Z=0$. The solid nodes are located in the grid structure as in Fig. 2.3.6.4. These node temperatures represent the volume average temperature for that node's cell and not necessarily the temperature at that location. For instance, the nodes at $Z=0$ and $Z=Z_{max}$ represent the volume average temperature for the whole cell and yet are located on the edge of the cell. Figure 2.3.6.4 also shows the stepped expansion near Z_{max} that is used to model the coolant channel collector in the lower reflector and the core support block coolant channel modifications which are shown in Fig. 2.3.6.5 as they appear in a typical HTGR. The multitude of converging coolant channels of Fig. 2.3.6.4 is still modeled as only one channel in the coolant calculations. When the coolant channels merge into fewer, larger channels, the model channel must be modified to maintain the correct heat transfer area (surface area of the channels), flow areas, and diameters. The area change for each expansion in the single channel model is proportional to the total flow and surface area change in the HTGR's refueling regions. The diameter used for the Reynolds number and heat transfer coefficient calculations is the characteristic diameter in the HTGR at the axial location.

The cell coolant temperature calculations are performed in the same manner for normal downflow (positive Z direction) or reversed flow (upflow or negative Z direction) in that the cell inlet temperature is the next upstream cell exit temperature. The calculational technique is marched in the negative Z direction for reverse flow and the positive Z direction for normal flow (downflow). The channel inlet temperatures for the normal (all downflow) case is the upper plenum temperature supplied by functional input values or results of CACS calculations in the ACHERONS package of subroutines. For the mixed flow case (i.e., some regions have upflow and some downflow), the inlet temperatures are the upper or lower plenum temperatures as supplied by the flow redistribution calculations. These temperatures are the mass average temperatures considering all of the flow sources and assuming a well mixed plenum. The same calculational technique is used for both the steady-state and transient analysis.

All coolant properties (conductivity, viscosity, and specific heat) and the heat transfer coefficient are calculated internal to the program and assume helium or helium-air mixtures (see section 2.3.11). The heat transfer coefficient is evaluated as in Eqs. (2.3.6.5), (2.3.6.6), and (2.3.6.7).

Fig. 2.3.6.4. Coolant Channel Description with Sample Node Locations

Fig. 2.3.6.5. Sample Coolant Channel Modification in Lower Reflector and Core Support Block

$$h = 0.656 \left(\frac{G C_p}{R_n^{2/3}} \right) \quad R_n < 2100 \quad (2.3.6.5)$$

$$h = 0.004 G C_p + (14.9 \frac{k}{D} - 0.004 G C_p) \left(\frac{R_n - 2100}{1900} \right) \quad 2100 \leq R_n \leq 4000 \quad (2.3.6.6)$$

$$h = 0.0196 R_n^{0.8} \frac{k}{D} \quad R > 4000 \quad (2.3.6.7)$$

where

$$R_n = \text{Reynolds number} = \frac{GD}{\mu}$$

D = reference diameter

k = gas conductivity

Equations (2.3.6.5) and (2.3.6.7) are taken from [Ref. 13]. Equation (2.3.6.6) represents a continuous transition from laminar to turbulence. The derivatives are not continuous at the match points for Eqs. (2.3.6.5) through (2.3.6.7), however, and improvements in the code could include smoothing this function at the match points.

In the transient temperature calculations, the coolant inlet properties and the solid temperatures are at the t_{n+1} time. This time level is used since the coolant temperature calculational technique is a steady-state solution and the coolant temperatures at the time t_{n+1} do not depend on information from any previous time level.

2.3.7. Flow Redistribution Technique for Transient Analysis

In the analysis of the thermal and coolant flow transients encountered in the reactor core, it is necessary to have a flow redistribution technique capable of handling both a normal flow condition where flow is down in all refueling regions and a reverse flow condition where flow is up in one or more refueling regions. A reverse flow condition occurs at very low or zero coolant flow situations when buoyant forces become significant causing the flow to reverse direction and flow up in some refueling regions. In HEXERE12 this transient flow redistribution technique is contained in subroutine CORFLO along with several smaller auxiliary subroutines.

Due to the large heat capacity of the graphite in the reactor core and the relatively small heat capacity of the coolant, the coolant temperatures are capable of changing much more rapidly than are the temperatures of the solid portion of the core. Because of this difference in the rate at which the solid and coolant temperatures can undergo a transient, the coolant temperature, and hence the coolant flow redistribution calculations, essentially follow the solid temperature transient. That is to say that at any particular time the coolant temperatures and coolant flow distribution are influenced strongly by the solid temperatures at that time and not by conditions at any previous time. Advantage is taken of this fact in order to simplify the flow redistribution calculation technique, and a steady-state calculation rather than a transient one is used in CORFLO. Each time CORFLO is called, it is supplied with the solid and

coolant temperatures and the total coolant flow rate. CORFLO uses this information to determine the flow in each refueling region and the upper and lower plenum mixed mean temperatures.

When CORFLO is called, it distributes the flow among the refueling regions such that (1) the pressure change across all refueling regions is the same within specified limits and (2) the total flow across the reactor agrees within specified limits with the given flow for the current time. The calculational procedure used in CORFLO is outlined below.

1. An initial estimate of pressure change across the core is made. (A positive pressure change indicates a pressure rise from the top to the bottom of the core and a negative pressure change indicates a pressure drop.) On the first call to CORFLO the initial estimate of pressure change across the core is arbitrarily assumed to be zero to start the iterative procedure. On all successive calls to CORFLO the initial estimate of pressure change across the core is assigned the value to which it converged on the previous call to CORFLO. From this estimate, an average pressure in the reactor can also be calculated.
2. The density head in each refueling region is calculated based on the coolant temperatures in each region and the average pressure in the reactor. This calculation is done according to Eq. (2.3.7.1).

$$h_{d,i} = \sum_{k=1}^n \frac{(\Delta Z_k)(P_{avg})}{T_{i,k} R} \quad (2.3.7.1)$$

where

$h_{d,i}$ = density head in refueling region i ,

n = number of axial coolant node levels,

ΔZ_k = axial height of coolant passage associated with coolant node level k ,

P_{avg} = average pressure in the core,

$T_{i,k}$ = coolant temperature associated with node level k in refueling region i ,

R = gas constant.

On successive iterations during the same call to CORFLO, the coolant temperatures do not change. Therefore, the density heads do not need to be entirely recalculated; they can be determined through modification to reflect the new average pressure in the reactor. This modification is given by Eq. (2.3.7.2).

$$h_{d,i} = h_{d,i} \left(\frac{P_{avgnew}}{P_{avgold}} \right) \quad (2.3.7.2)$$

where

P_{avgnew} = new estimate of core average pressure,

P_{avgold} = previous estimate of core average pressure.

3. The flow direction in each refueling region is determined, based on the latest estimate of pressure change across the core and density head in each region. Subtracting the density head for a particular refueling region from the pressure change across the core

gives the pressure change due to flow for that region. If the pressure change due to flow for a region is negative, the flow is down in that region; if it is positive, the flow is up.

4. The flow in each refueling region is determined so that the pressure change across each region agrees with the latest estimate of pressure change across the core. The iterative procedure for determining the flow in a particular region is as follows:
 - (a) Using the latest estimate of flow in the region, all Reynolds numbers dependent loss coefficients (friction, expansions, contractions) are calculated. The orifice loss coefficient is assumed to be independent of Reynolds number and the value determined in the steady-state flow redistribution analysis is used. An acceleration loss coefficient is also determined from the latest estimate of coolant inlet and outlet temperatures. A friction loss coefficient is defined for each channel such that the pressure change due to friction for a channel is the loss coefficient for that channel times the square of the mass flow rate in that channel. That is

$$\Delta P_{\text{fric}_i} = K_{\text{fric}_i} W_i^2 \quad (2.3.7.3)$$

where

ΔP_{fric_i} = frictional pressure change in channel i,

K_{fric_i} = friction loss coefficient for channel i,

W_i = mass flow rate in channel i.

The friction loss coefficient is determined by summing the pressure changes for all of the increments along the channel and factoring out the mass flow rate squared. This is shown in Eq. (2.3.7.4). The term in parentheses is the friction loss coefficient for the channel.

$$\Delta P_{\text{fric}_i} = \left(\sum_{k=1}^n \frac{1}{2g_c \rho_k A_k^2} f_k \frac{L_k}{D_k} \right) W_i^2 \quad (2.3.7.4)$$

where

n = number of axial nodes,

ρ_k = density at axial location k in refueling region i,

A_k = flow area at axial location k in refueling region i,

f_k = friction factor,

L_k = length associated with node level k in region i,

D_k = reference diameter.

The friction factor is Reynolds number dependent and the equations for determining it in the laminar, transitional, and turbulent regions are given by Eqs. (2.3.7.5), (2.3.7.6), and (2.3.7.7), respectively.

$$f = \frac{64}{R_n} \quad R_n < 2100 \quad (2.3.7.5)$$

$$f = 5.361 \times 10^{-6} R_n^{0.94896} \quad 2100 \leq R_n \leq 3000 \quad (2.3.7.6)$$

$$f = \frac{0.3164}{R_n^{0.25}} \quad R_n > 3000 \quad (2.3.7.7)$$

where

R_n = Reynolds number.

Equations (2.3.7.5) and (2.3.7.7) are derived from Ref. 14 by conversion to a Darcy friction factor. Equation (2.3.7.6) is a straight line on a log-log plot such that the friction factor is continuous in the transition from laminar to turbulent flow. Although Eq. (2.3.7.6) does match Eqs. (2.3.7.5) and (2.3.7.7) at the end points, the derivatives do not.

The value for the portion of expansion loss coefficient due to irreversible losses for different Reynolds numbers is given by Eqs. (2.3.7.8) - (2.3.7.11).

$$K_{irr} = 1 - 2.667X + X^2 \quad R_n \leq 1900 \quad (2.3.7.8)$$

$$K_{irr} = 1 - 2.14X + X^2 \quad R_n = 2000 \quad (2.3.7.9)$$

$$K_{irr} = 1 - 2.10X + X^2 \quad R_n = 1 \times 10^4 \quad (2.3.7.10)$$

$$K_{irr} = 1 - 2.0X + X^2 \quad R_n \geq 1 \times 10^8 \quad (2.3.7.11)$$

where

$$X = \frac{A_{up}}{A_{dn}}$$

A_{up} = flow area upstream of expansion,

A_{dn} = flow area downstream of expansion.

The expansion loss coefficient is determined by Eq. (2.3.7.8) for values of Reynolds number less than or equal to 1900 and by Eq. (2.3.7.11) for values of Reynolds number greater than or equal to 1×10^8 . For Reynolds number between 1900 and 1×10^8 , the expansion loss coefficient is determined by linear interpolation between values of Reynolds number at which a value is known. These known points are at Reynolds numbers of 1900, 2000, 1×10^4 , and 1×10^8 . The values of expansion loss coefficient at these points are given by Eqs.

(2.3.7.8) - (2.3.7.11). In addition to irreversible losses at an expansion, there is also a pressure change due to the change in flow area. The loss coefficient for the total pressure change (i.e., irreversible losses plus change due to flow area change) is given by Eq. (2.3.7.12).

$$K_e = 1 - X^2 - K_{irr} \quad (2.3.7.12)$$

Equations (2.3.7.8) - (2.3.7.11) are taken from Ref. 15.

The contraction loss coefficient is determined in a manner similar to the expansion loss coefficient. The irreversible part for varying Reynolds numbers is given by Eqs. (2.3.7.13) - (2.3.7.16).

$$K_{irr} = 1.08 - 0.4X \quad R_n \leq 1900 \quad (2.3.7.13)$$

$$K_{irr} = 0.56 - 0.4X \quad R_n = 2000 \quad (2.3.7.14)$$

$$K_{irr} = 0.5 - 0.4X \quad R_n = 1 \times 10^4 \quad (2.3.7.15)$$

$$K_{irr} = 0.4 - 0.4X \quad R_n \geq 1 \times 10^8 \quad (2.3.7.16)$$

The total loss coefficient is given by Eq. (2.3.7.17).

$$K_c = 1 - X^2 + K_{irr} \quad (2.3.7.17)$$

Equations (2.3.7.13) - (2.3.7.16) are taken from Ref. 16.

The acceleration loss coefficient is given by Eq. (2.3.7.18).

$$K_{acc} = \left(\frac{1}{\rho_{ex}} - \frac{1}{\rho_{in}} \right) \frac{1}{A^2 g_c} \quad (2.3.7.18)$$

where

ρ_{ex} = coolant density leaving the channel,

ρ_{in} = coolant density entering the channel,

A = coolant channel flow area.

- (b) Knowing the loss coefficients, the flow necessary to give the desired pressure change across the core is easily calculated by Eq. (2.3.7.19).

$$W_{new} = \left(\frac{\pm \Delta P_{flow}}{K_f + K_{acc} + K_{orf} - \Sigma K_e + \Sigma K_c} \right)^{0.5} \quad (2.3.7.19)$$

where

ΔP_{flow} = pressure change due to flow,

K_f = friction loss coefficient,

K_{acc} = acceleration loss coefficient,

K_{orf} = orifice loss coefficient,

K_e = an expansion loss coefficient,

K_c = a contraction loss coefficient.

The sign in front of ΔP_{flow} is determined by the flow direction. For downflow it is negative and for upflow it is positive.

- (c) If two successive iterations differ by less than a prescribed amount, the flow is assumed to be converged in that region. Otherwise, the program returns to (a) and the procedure is repeated until the convergence criterion is satisfied.

The iterative procedure just outlined is repeated for all refueling regions. Both upflow and downflow channels are handled in an identical manner except that different loss coefficients are used for the two cases. Since an expansion in one direction is a contraction when flow is reversed and vice versa, the orifice loss coefficient is assumed to be the same regardless of flow direction.

5. The flow is summed for all refueling regions and compared to the prescribed flow. If the calculated and prescribed flows differ by less than a specified convergence criterion, the calculations end. Otherwise, a new estimate of pressure change across the core is made and the program returns to (2) and continues to iterate until convergence is obtained.
6. Based on the flows calculated by CORFLO and coolant temperatures passed to CORFLO, upper and lower plenum mixed mean temperatures are calculated. The upper plenum mixed mean temperature is given by Eq. (2.3.7.20) and the lower plenum mixed mean temperature by Eq. (2.3.7.21).

$$T_{up} = \frac{\sum_{i_{up}} T_{ex,i} W_i + T_m W_{tot}}{\sum_{i_{up}} W_i + W_{tot}} \quad (2.3.7.20)$$

where

T_{up} = upper plenum mixed mean temperature,

$\sum_{i_{up}}$ = summation over all upflow channels,

$T_{ex,i}$ = coolant exit temperature for region i,

W_i = coolant flow rate for region i,

T_m = temperature of circulating coolant entering upper plenum,

W_{tot} = total coolant flow.

$$T_l = \frac{\sum_{i_{dn}} T_{ex_i} W_i}{\sum_{i_{dn}} W_i} \quad (2.3.7.21)$$

where

T_l = lower plenum mixed mean temperature,

$\sum_{i_{dn}}$ = summation over all downflow channels.

2.3.8. Flow Redistribution for Steady-State Analysis

The initial conditions necessary to start an HTGR accident transient analysis with coolant flow must include complete solid and coolant temperature descriptions, coolant channel flow rates, and refueling region orifice area settings and coefficients. The great amount of details necessary made it advantageous to incorporate a steady-state, coupled convection-conduction analysis technique to HEXEREI2. Since the starting conditions on coolant flow rate inlet temperature and pressure are well defined in most accident analyses, there was no need to incorporate a model for the steam generators, main circulators, or other portions of the reactor system. In the steady-state analysis, only the HTGR core structure is considered and the coolant inlet properties are as specified in the input data.

The flow redistribution technique is further simplified by allowing only two options in the assumed orifice setting technique. In the first option, the core coolant exit temperatures are maintained equal for all refueling regions. Since the coolant exit temperature is not a function of core pressure drop but only of mass flow rate, the coolant channel pressure drop does not need to be considered in the steady-state flow redistribution technique. The individual channel flow rates need only be adjusted by the ratio of the coolant average exit temperature over the channel's exit temperature.

Of course, the redistribution technique must interact with the solid and coolant steady-state iterative calculations until all temperatures and flow rates converge.

The second option is to specify each channel's flow rate in the input data, and then execute the temperature calculations until converged. This option allows the user to utilize a control function other than uniform exit temperature for a steady-state analysis.

After the mass flow rates, solid and coolant temperatures are determined by either of the above procedures, the orifice, loss coefficients, and area settings are determined. The orifices in the HTGRs are constrained to have both a minimum and maximum loss coefficient. In HEXEREI2, the refueling region that achieves maximum flow from the above techniques is assumed to have an orifice coefficient equal to the minimum [Ref. 16]. Since the mass flows and axial temperature distribution are known for all regions, the pressure drop across the high flow region can be calculated and produces the overall core pressure drop. The only pressure drops not known for the rest of the regions are their orifice pressure drops. By subtracting the calculated incremental pressure drops from the core pressure drop, the orifice pressure is found. From the relationship in Eq. (2.3.8.1), the orifice coefficients are obtained. The orifice areas can be obtained from Ref. 16.

$$k_o = \frac{\Delta P}{\dot{m}^2} 2g_c A_o P \quad (2.3.8.1)$$

where

A_o = orifice cylinder area

ΔP = pressure drop

$\dot{m}$ = mass flow rate.

2.3.9. Modeling Technique for HTGR Auxiliary Heat Exchangers

The method discussed in this section is an extension of work by J. R. Tallackson reported in Ref. 17.

2.3.9.1 General

The heat exchangers discussed below represent the 2000 and 3000 megawatt exchangers. Although they differ in size and tube material, they have the same relationship between flow direction and general tube geometry. Because of their similarities, the two exchangers are modeled with the same code utilizing different input parameters. The Fort St. Vrain steam generator is discussed separately (section 2.3.9.2).

2.3.9.1a General Description of HEXEREI2 Heat Exchanger Model

The heat exchangers used in GAC's HTGRs are all constructed of helically coiled tubes with internal water (or steam) flow counter direction to the gas flow. The tube bank as shown in Fig. 2.3.9.1 is an assembly of concentric, cylinder-like tube sheets of helically coiled tubes with different helix angles for each cylinder. The tube bundle inlet and outlet headers are neglected in HEXEREI2. In all of the HTGR heat exchangers modeled to date, the water tubes have approximately the same length from the top to the bottom of the heat exchanger bundle.

Since adjacent cylindrical tube sheets may have different helix angle (angle tube helix makes with a plane to the perpendicular center line of the heat exchangers), a cross sectional drawing of a portion of the tube bundle (Fig. 2.3.9.2) shows what appears to be traditional triangular tube spacing in some areas, rectangular in others, and a mixture in the rest. This configuration does not exactly duplicate those of the accepted tube bundle heat transfer correlations [Ref. 18]. An approximate technique of apportioning the effects on the heat transfer correlation from the triangular and rectangular portion of the heat exchanger bundles is utilized.

2.3.9.1b Numerical Procedure

The gas and water temperatures are calculated by an incremental heat balance starting at the helium inlet (hot) end of the heat exchanger and marching to the exit (cold) end. For the HTGR heat exchangers that are counter flow, the water exit temperature is guessed and the change in heat for the gas and water are balanced for a small section of the heat exchanger with an iterative technique. The values of exit helium and inlet water temperatures are then used as the starting conditions for the next section in the gas flow direction. When the end of the heat exchanger is reached, the calculated water inlet temperature is compared to the input water exit temperature; if necessary, the water exit temperature is adjusted, and the section by

32-RS-73140 A

Fig. 2.3.9.1. Core Auxiliary Cooling Loop Arrangement

Fig. 2.3.9.2. Heat Exchanger Tubing Distribution in R-Z Plane

section incremental heat balance is repeated until the calculated water exit temperature agrees with the input value. If any portion of the heat exchanger has parallel flow, then the above technique is modified (see section 2.3.9.2).

The above technique is a steady-state technique in that no consideration in the heat balance has been given to the transient heat capacity of the tube wall material or the fluids. Figure 2.3.9.3 depicts the iterative and marching technique.

The incremental heat balance is shown in Eq. (2.3.9.1).

$$\dot{q}_{Hi} = \dot{m}_2 C_{p2}(T_{E2} - T_{I2})_i = \dot{m}_1 C_{p1}(T_{E1} - T_{I1})_i \quad (2.3.9.1)$$

where

$\dot{q}_{Hi}$ = incremental heat duty

T_E = fluid exit temperature

T_I = fluid inlet temperature

$\dot{m}$ = fluid mass flow rate

C_p = fluid specific heat

subscript

1 - gas parameters

2 - water parameters

i - section index

Since the geometry of the heat exchanger is constant along its axis, it can be modeled, for the purpose of calculations, as a simple concentric shell and tube arrangement with the appropriate heat transfer coefficient applied to each surface tube. This model with sample sections is shown in Fig. 2.3.9.4. For this type of model

$$\dot{q}_{Hi} = U_i A_i (T_{1i} - T_{2i}) \quad (2.3.9.2)$$

where

A_i = incremental heat transfer area

U_i = section overall heat transfer coefficient

T = fluid bulk temperature

The overall heat transfer coefficient, U , based on the outside area is

$$\frac{1}{U} = \frac{1}{h_o} + \frac{L}{k_w D_{lm}} + \frac{D_o}{h_i D_i} \quad (2.3.9.3)$$

Fig. 2.3.9.3. Graphical Representation of Heat Exchanger Calculations

where

h_o = outside heat transfer coefficient

h_i = inside heat transfer coefficient

D_o = outside diameter of tube

D_i = inside diameter of tube

L = tube wall thickness

$D_{lm} = (D_o - D_i) / \ln(D_o / D_i)$

k_w = conductivity of tube wall

Fig. 2.3.9.4. Heat Exchanger Model with Sample Sections

The convective heat transfer coefficient, h_i , on the water side of the tube is determined by the Dittus-Boelter correlation.

$$Nu_i = 0.023 Re_i^{0.8} Pr^{0.4} \quad (2.3.9.4)$$

where

$$Nu_i = \frac{h_i D_i}{k} = \text{Nusselt number (dimensionless)}$$

$$Re_i = \frac{D_i G}{\mu} = \text{Reynolds number (dimensionless)}$$

$$Pr = \frac{C_p \mu}{k} = \text{Prandtl number (dimensionless)}$$

D_i = inside diameter of tube

G = mass flux of water

k = conductivity of water

μ = viscosity of water

C_p = heat capacity of water

The properties of water are evaluated at the bulk water temperature.

The convective heat transfer coefficient on the gas (helium or helium-air mixture) side of the tubes, h_0 , is determined by a modified Grimson correlation assuming that the tube bundle is a combination of both in-line and staggered tube arrangements [Ref. 17].

$$Nu_0 = X_1 F_1 C_1 Re_0^m Pr^{0.33} + X_2 F_2 C_2 Re_0^n Pr^{0.33} \quad (2.3.9.5)$$

where

X_1, X_2 = fraction of tubes that are staggered and in-line, respectively.

C_1, C_2 = leading constants for staggered and in-line tubes, respectively, that are functions of lateral and longitudinal spacing.

m, n = exponents for staggered and in-line tubes, respectively, that are functions of lateral and longitudinal tube spacing.

F_1, F_2 = 'arrangement factors,' function of lateral and longitudinal tube spacing and Reynolds number.

The properties on the gas side are evaluated at the gas-side film temperature. In the computer program, the coefficients F and C are combined into a single term and are determined from input data. The Nusselt number coefficients are obtained from a method developed by J. P. Sanders using the best fit to the statistical trend for the Grimson coefficients [Ref. 15]. This fit was evaluated by D. F. Hoppes at ORNL against recent proprietary data from the European community; the results were presented in an internal ORNL report titled "Evaluation of Recent Data for Gas-Side Heat Transfer Coefficients for Tube Bundles." In Table 9 of that report, the necessary equations and constants needed to calculate the Nusselt number coefficients are presented. Subroutine TUBE stores these constants and uses the relationships from the report to obtain the Nusselt number coefficients. If the transverse or longitudinal distances are not specified exactly in the tables, the subroutine uses

an interpolation procedure to obtain those constants necessary for determining the Nusselt number coefficients. The heat loss to the walls of the CACS cavity is neglected in this analysis technique.

The pressure loss in the HTGR heat exchangers is determined in a manner similar to the heat transfer correlation calculations by calculating the friction factor for the staggered and in-line tubes, and weighting each with the fraction of each tube configuration in the heat exchanger bundle.

The empirical friction factor, f' , was calculated using Jakob's approach for Reynolds number greater than 1000 [Ref. 19], i.e.,

$$f'_1 = 0.25 + \frac{.118}{\left(\frac{S_T - D_0}{D_0}\right)^{1.08}} \left(\frac{G_{max} D_0}{N_b}\right)^{-1.6} \quad (2.3.9.6)$$

for staggered tubes, and

$$f'_2 = .044 + \frac{.08 S_L / D_0}{\left(\frac{S_T - D_0}{D_0}\right)^{.43 + 1.13 D_0 / S_L}} \left(\frac{G_{max} D_0}{N_b}\right)^{-1.5} \quad (2.3.9.7)$$

for in-line tubes. Since the tubes are arranged N1 in-line and N2 staggered, the friction factor was assumed to be

$$f = X_2 f'_2 + X_1 f'_1 \quad (2.3.9.8)$$

and the pressure drop was calculated by

$$\Delta P = \frac{f G_{max}^2 N}{(2.09 \times 10^8)} \left(\frac{\mu_s}{\mu_b}\right)^{.14} \quad (2.3.9.9)$$

where

ΔP = pressure drop (lb/ft²)

G_{max} = mass velocity at minimum free flow area

ρ = density (lb_m/ft³)

N = number of transverse rows

μ_s = viscosity at tube surface temperature,

μ_b = viscosity at helium bulk temperature

Since the heat exchanger is divided into several sections, the change in pressure is calculated for each section based on the average properties for that section. The total pressure drop is then obtained by adding the sectional pressure drops.

In the 3000 and 2000 megawatt HTGR designs, a portion of the flow entering the upper plenum bypasses the core by flowing through the gas annulus surrounding the side reflector. At present, this flow is accounted for in the full loop analysis by assuming no temperature change through the gas annulus, and that the bypass flow is well-mixed with the core exit flow before entering the CACS. Future code development could include the calculations of the heat transfer in the gas annulus in the same manner the coolant channel heat transfer is calculated.

2.3.9.2 Fort St. Vrain Steam Generator

Coding was developed for HEXEREI2 to model the 3000 and 2000 megawatt HTGR CACS heat exchangers and the Fort St. Vrain steam generator when used in auxiliary heat removal. Thus, by using different routines, the three heat exchangers could be modeled in the same HEXEREI2 code.

The two CACS heat exchangers are modeled with one routine where all of the model parameters are obtained from input data (see section 3.6.16f). These modeling parameters are the same as above.

Since the Fort St. Vrain steam generator requires a considerably modified technique, the modeling parameters are contained within the routine required for this analysis. Thus, they are removed from normal user control.

The Fort St. Vrain steam generator differs from the previous general description of CACS heat exchangers in that it contains four separate exchangers that see the gas flow sequentially. Upon leaving the reactor core, the primary coolant of the Fort St. Vrain flows first through a reheater then into Superheater 2 and finally into the steam generator. The steam generator consists of three sections. The sections, as they are encountered by the primary coolant, are: (1) Superheater 1, (2) the evaporator, and (3) the economizer. The flow is countercurrent in all the heat exchangers with the exception of Superheater 2, which has parallel flow. Fig. 2.3.9.5 presents a simplified diagram of the primary (gas) and secondary (water) flow systems.

In developing a model of the heat transfer system, provisions had to be made for the varying physical characteristics of each heat exchanger. This was accomplished by developing a model which divided the steam generator into three distinct sections. The model neglected the reheater since the water or steam flow through the reheater had been negligible in the accident transients analyzed to date.

The code incorporates the physical characteristics shown in Ref. 20 and repeated in Table 2.2. Each of the four exchanger sections was divided into multiple subsections to improve accuracy and convergence. The iterative procedure involved is to estimate the outlet water temperature at Superheater 1 and back-calculate the inlet water temperature to the economizer. The heat exchanger calculations are assumed to converge when the calculated initial temperature is within a sufficiently small tolerance of the known value. The calculation of parallel flow (Superheater 2) effect on primary and secondary coolant temperatures was fairly straightforward since Superheater 1 water exit temperature was assumed. The procedure is to estimate a Superheater 2 inlet temperature and, using this, calculate the primary coolant exit temperature from Superheater 2. This exit temperature was then used in the calculations for the rest of the exchangers. Since the other exchangers are sequential counterflow exchangers, the technique used is very similar to the CACS heat exchanger calculations except for the change in physical properties for each exchanger.

2.3.10. CACS Circulator Simulation

Both the CACS and the main circulators in GAC's HTGR system are single stage (rotor-stator) axial compressors. The main circulator is sized to provide the design flow rate (full power flow rate) at the design pressure. The CACS circulator is designed to provide a much smaller flow rate at a much lower system pressure. The CACS circulator in the 2000/3000 HTGRs is powered by a synchronous electric motor rather than a steam turbine as in the main circulator. In general, these circulators can be approximated with an ideal compressor, since compressible and viscous effects are negligible.

Fig. 2.3.9.5. Fort St. Vrain Steam Generator - Simplified Description

Table 2.2. Steam Generator and Reheater Physical Characteristics

PART I	SINGLE MODULE DATA		
	EES (Economizer- Evaporator- Superheater 1)	SH2 (Superheater 2)	RH (Reheater)
A. Effective heat transfer area without leads, ft ²			
Installed	2480	850	440
Ideal	2120	704	388
B. Superficial gas flow area, ft ² . Area occupied by:	18.634	15.569	18.201
Tubing	11.657	9.367	11.636
Support structure	0.435	0.420	0.442
C. Effective transverse tube spacing (pitch), in.	1.42	1.48	1.575
D. Effective longitudinal tube spacing (pitch), in.	1.47	1.44	1.575
E. Tube outside diameter, in.	1.00	1.00	1.125
F. Tube wall thickness, in.	.138/.138/.225	.205	.140
G. Tube material	SA-213 T2/T22/T22	SB-163-GR2 NI-FE-CR	SB-163-GR2 NI-FE-CR
H. Effective tube length, ft			
Installed	175.	60.	15.7
Ideal	149.99*	49.81	15.7
I. Cross-sectional free flow	6.542	5.782	6.123

* Ideal Econ. Length = 76.81 ft
 Ideal Evap. 1 Length = 25.51 ft
 Ideal Evap. 2 Length = 23.38 ft
 Ideal Superheater 1 Length = 24.29 ft
 TOTAL = 149.99 ft

Division of EES *installed* surface not readily available. Recommend using same percentage division as for ideal if needed for analysis.

Ideal compressors satisfy the following similarity equations:

$$\frac{\phi/\phi_o}{n} = \text{constant} \quad (2.3.10.1)$$

$$\frac{\psi/\psi_o}{n^2} = \text{constant} \quad (2.3.10.2)$$

where

n = circulator speed RPM

ϕ = corrected mass flow rate = $\frac{GD}{\mu}$

ψ = $\Delta P/P$

m = mass flow rate lb/sec

ΔP = pressure rise across circulator

P = coolant pressure

T = gas temperature

θ = T/T_o

R = coolant gas constant

δ = P/P_o

subscript o refers to design conditions.

The pressure drop for a particular corrected speed and mass flow rate can be calculated from this information if the pressure drops and flow rates for a constant corrected speed line are known. This information is generally provided in the form of a normalized compressor map (Fig. 2.3.10.1) with the normalizing parameters reported.

The corrected speed can be determined by either the required flow rate ratioed to the design flow rate as in Eq. (2.3.10.3) or an actual speed required by a control function, adjusted to a corrected speed as in Eq. (2.3.10.4).

$$n = n_o \sqrt{\theta} \sqrt{\frac{R}{R_o}} \quad (2.3.10.3)$$

$$N_c = \frac{n}{\sqrt{\theta} \sqrt{\frac{R}{R_o}}} \quad (2.3.10.4)$$

Once the speed and circulator pressure rise have been determined, the shaft power is determined from Eq. (2.3.10.5).

$$W_{shaft} = \frac{1}{\eta} \frac{RT}{(k-1)/k} \left[\left(\frac{P_e}{P_i} \right)^{(k-1)/k} - 1 \right] \quad (2.3.10.5)$$

$$= \frac{1}{\eta} \frac{RT}{(k-1)/k} \left[\left(\frac{\Delta P}{P_i} + 1 \right)^{(k-1)/k} - 1 \right]$$

Fig. 2.3.10.1. Sample Circulator Performance Map

where

$\eta = 0.75$, compressor efficiency

$\Delta P = P_e - P_i$, circulator pressure rise

P_e = exit pressure

P_i = inlet pressure

Because of complexity of the primary coolant system, two versions of the circulator numerical technique were developed. The first is simply a performance evaluation technique because it calculates the circulator performance and only reports it. This technique does not provide the results of its calculation to the rest of HEXEREI2. In order to proceed with the transient, HEXEREI2 must be provided with time-dependent coolant flow rate, pressure and temperature at the inlet to the core. In this simple technique, called an 'open loop' analysis, the input data must supply this information. Using the 'open loop' technique, transient analysis can be performed without having to describe fully the whole primary coolant system. This technique also enabled evaluation of HEXEREI2 where complete description of the primary coolant system and all modeling assumptions of a comparison code were not available. In comparison of code results, one difference in modeling technique might produce an accumulative error in a transient analysis preventing a valid comparison of the components in the analysis. The 'open loop' analysis helped reduced this error.

The second technique, termed a 'closed loop' analysis, supplies the rest of HEXEREI2 with the sufficient core coolant inlet values to enable a transient analysis without any time-dependent parameters being supplied. The only exception to the above statement is the coolant molecular weight. At present, this value must be input as a function of time. The inclusion of a previously developed program, INGRES [Ref. 21], would eliminate this requirement.

The 'closed loop' analysis is much more complicated, hampered by possible convergence problems, and requires considerable detail of the circulator. Very briefly, the 'closed loop' utilizes reestimation techniques based on the ideal compressor relationships, ideal counter flow heat exchanger with constant heat transfer coefficients, and a system pressure loss correlation with a constant loss coefficient to iteratively find a circulator speed that satisfies all of the constraints of the flow system. In the 2000/3000 mw HTGR, these constraints include: 1) a limit on the water temperature in the CACS heat exchanger bundle, 2) circulator motor speed limit, and 3) circulator motor torque and speed envelope. The circulators must also provide sufficient pressure rise to overcome the pressure drop in the rest of the flow system. If the circulator is required, due to control functions to produce more or less pressure rise, then the system flow as determined by the 'closed loop' technique will accelerate or decelerate as necessary.

In the Ft. St. Vrain HTGR, the main steam generator and circulator are used to provide auxiliary heat removal. The circulator can be powered by either the normal steam turbine using auxiliary steam or a water turbine (Pelton wheel) with various sources of pressurized water. When powered by the Pelton wheel, a fixed water pressure is assumed and the circulator torque and speed are iteratively solved based on the Pelton wheel's capability. It is also possible to fix the circulator speed during an analysis.

2.3.11. Coolant Properties

The expected types of coolant present in the HTGR during normal and accident operation are incorporated into HEXEREI2 by the properties below. The gas mixture properties are those developed by J. R. Tallackson [Ref. 22] and provide for air ingress into the PCR. After contact with the hot graphite, the air-helium mixture is assumed to produce a mixture of helium, carbon monoxide, hydrogen gas (from the moisture in the air), and nitrogen.

The temperature-dependent properties of water used in the calculations are determined from the following expressions [Ref. 23].

Viscosity

$$\mu = \frac{378.4}{T^{1.81}} \quad (2.3.11.6)$$

Conductivity

$$k = 0.31 + 6.91 \times 10^{-4}T - 1.48 \times 10^{-8}T^2 \quad (2.3.11.7)$$

Specific Heat

$$C = 1.035 - 4.55 \times 10^{-4}T + 1.38 \times 10^{-6}T^2 \quad (2.3.11.8)$$

where

T = temperature (°F)

If the programmed coolant properties are to be used, then the analysis must be performed in British (engineering) units. The simple coding change to utilize SI units has not been performed since all analyses have been compared to GAC's results reported in British units.

Viscosity

$$\mu_{He} = 5.942 \times 10^{-4} T_a^{.71} \quad (2.3.11.1)$$

where

μ = viscosity (lb_m/hr-ft)

T_a = absolute temperature (°R)

Conductivity

$$K_{mix} = \left[\frac{.0399}{(.27716 + X_1)} - .3196 \times 10^{-2} + .07014 X_1 - .15767 X_1^2 + .08618 X_1^3 \right] \frac{T_a}{1061} \quad (2.3.11.2)$$

where

$$X_1 = \frac{AMW - AMW_2}{AMW_1 - AMW_2}$$

and

AMW = molecular weight of mixture of air and helium

AMW₁ = molecular weight of air

AMW₂ = molecular weight of helium

k = conductivity (Btu/hr-ft-°F)

T_a = absolute temperature (°R)

Specific Heat

$$C_{air} = 0.2438 - 0.29401 \times 10^{-4} T_a + 0.45436 \times 10^{-7} T_a^2 - 0.1120 \times 10^{-10} T_a^3 \quad (2.3.11.3)$$

$$C_{He} = 1.2425 \quad (2.3.11.4)$$

$$C_{mix} = g_{air} C_{air} + g_{He} C_{He} \quad (2.3.11.5)$$

where

C = heat capacity (Btu/lb_m-°F)

g = mass fraction of air or helium

3 DESCRIPTION OF INPUT AND OUTPUT

3.1 General

This section is designed to guide the user through the steps necessary to solve a heat transfer problem on HEXEREI2. First, a set of instructions is presented to aid the user in representing the geometrical appearance of the problem with lattice points. Then the functions used by the code to define the material parameters are given. A detailed discussion of the input cards and a brief outline of the input data for the benefit of the user who is familiar with the data preparation are presented, followed by a brief description of the normal output. In preparing the input data, any consistent set of units may be used in the HEXEREI2 program except for problems involving radiation or convection to coolant. For radiation all temperature units must be in either degrees centigrade or Fahrenheit. Coolant calculations require temperature in degrees Fahrenheit. The units associated with the algorithms which appear in user-supplied functions must be consistent with those of the input data.

3.2 Regions

The basic HEXEREI2 model (see section 2.2) is a right circular cylinder. Many regions may comprise this basic model depending on the variation in material properties, heat generation, and boundary conditions desired. There are four basic rules governing region geometry.

- 1) Boundary planes must be parallel to the coordinate axes. Thus, all regions must be portions of a cylindrical solid (at present, they must also extend the full circumference of the cylindrical solid).
- 2) A region may contain only one material (however, many regions may contain the same material). The material in a region can be modified by the user-supplied subroutine for a special case applicable to the HTGR (see section 5).
- 3) A region may have up to one heat generation function (again, many regions may have the same heat generation function).
- 4) When a boundary condition is defined along the boundary of a region, it must apply over the boundary plane. Only one boundary is allowed on the boundary plane of a region but the same boundary may be applied to many regions.

The region configuration is independent of the nodal mesh with the exceptions stated in the following section. The nodal mesh within one region can be regular hexagonal only, cylindrical only, a mixture of interfacial and hexagonal, and or cylindrical (see section 2.2).

3.3 Lattice Arrangement

The second general requirement for describing the overall configuration is to construct the nodal spacing. In all but the hexagonal mesh, the node points are defined as intersections of lattice lines (planes). There are two classes of lattice lines: fine and gross. The gross lattice lines are input data and a gross lattice line must be specified at both region boundaries for each axis. Fine lattice lines, equally spaced, may appear between two consecutive gross lattice lines to create a finer mesh in the axial direction and in the cylindrical mesh portion of the nodal plane.

In the hexagonal mesh zone, the input gross lattice line and hexagonal pitch are used in a different manner to generate the lattice structure. Lattice lines parallel to the axial coordinate (nodal columns) are generated on a regular hexagonal pitch and the intersection of these lines with the axial planes, determined from the user's input, define the node points. The input gross lattice lines control the distribution of nodal columns in the following manner. The hexagonal mesh will really lie between the first two radial gross lattice lines (planes), constructed in such a manner that the interfacial cells will be outside of the second gross lattice line. The inner portion of the cylindrical mesh will lie on the first radial lattice line (fine or gross) outside of the second radial gross lattice line. The above procedure is intended to produce a mesh structure appropriate to HTGR analysis when the active core radius is input as the second radial gross lattice line.

The node points are numbered outward from the center of the nodal (axial) plane with the smallest z coordinate. When one nodal plane has been completed, the next consecutive node is the center of the next nodal plane.

3.4. Analytical and Tabular Functions

The analytical and tabular functions are built-in functions which may be used to aid in the description of some of the input parameters. An analytical function is defined by

$$F(v) = A_1 + A_2v + A_3v^2 + A_4\cos(A_5v) + A_6\exp(A_7v) + A_8\sin(A_9v) + A_{10}/\ln(A_{11}v) \quad (3.1)$$

A tabular function is defined by a set of ordered pairs, $(v_1, G(v_1)), (v_2, G(v_2)), \dots, (v_n, G(v_n))$, where the first element of a pair is the independent variable and the second is the corresponding value of the function. In order to evaluate the tabular function at some point, the program uses linear interpolation in the interval containing the point. The set of ordered pairs must be chosen so that the independent variable is arranged in ascending order or

$$v_1 < v_2 < v_3 < \dots < v_{n-1} < v_n. \quad (3.2)$$

If the function must be evaluated at some point outside of the defined domain of the function, then

$$G(v) = \begin{cases} G(v_1), & v < v_1 \\ G(v_n), & v > v_n \end{cases} \quad (3.3)$$

3.5. Input Parameters

The input parameters included in Table 3.1 must be defined by a function having the following form:

$$P(x,y,z,t,T) = P_0 \cdot f(x,y,z,t,T), \quad (3.4)$$

where P_0 is a constant factor and $f(x,y,z,t,T)$ may be a product of analytical and tabular functions, such as

$$f(x,y,z,t,T) = F_1(x) \cdot F_2(y) \cdot F_3(z) \cdot F_4(t) \cdot F_5(T) . \quad (3.5)$$

Table 3.1. Dependence of Input Parameters

Parameter	Function of					Related User-Supplied Subroutine
	x	y	z	t	T	
k				(x)	x	CONDIN
ρ				(x)	x	DENSITY
C_p				(x)	x	CPHEAT
T_o	x	x	x	(x)		INITTP
Q	x	x	x	x	x	HEATGN
T_b				x		BNDTMP
h_c	(x)	(x)	(x)	x	x	CONVTN
h_r	(x)	(x)	(x)	x	x	RADITN
h_n	(x)	(x)	(x)	x	x	NATCON
h_e	(x)	(x)	(x)	x	x	NCONEX
h_f	(x)	(x)	(x)	x	x	BNFLUX

NOTE: The x enclosed in parentheses, (x), indicates that although the parameter cannot be an explicit function of the indicated variable using normal input data, this variable is initialized in the respective user-supplied subroutine. Thus, the parameter can be a function of the indicated variable if it is so defined in its respective user-supplied subroutine.

In observing the definitions of the input parameters (see Table 3.1) which may be defined by Eq (3.4), it is noted that only the volumetric heat generation rate is a function of all the independent variables. Thus, if any variable is omitted from the definition of the parameter, then the corresponding factor is set equal to unity in Eq (3.5). The constant factor, P_0 , is part of the input data, and its value appears on the data card which is used to define the parameter. If one of the input parameters is to be a function of position, time, or temperature, then the appropriate index or indices, i, j, k, l or m , from Eq (3.5) are entered on a data card, too. If an index is positive, then it defines the number of the analytical function for the respective variable. If it is negative, then the absolute value of the index defines the number of the tabular function for the respective variable. If any of the defined factors for a parameter are omitted from the input data, then that particular factor is set equal to unity in Eq (3.5). If none of the factors are defined in the input, then that particular parameter is treated as a constant. If the value of P_0 is zero or is left blank on the data card and if the data indicates that the parameter is to be a function of position, time or temperature, then P_0 is set equal to unity in Eq (3.4). If the value of P_0 is zero or is left blank and if the data indicates that the parameter is not a function of position, time, or temperature, then P_0 is set equal to zero in Eq (3.4).

If the parameter cannot be defined by a product of analytical and tabular functions as indicated in Eq (3.5), then the user may supply his own function to evaluate Eq (3.4). Table 3.1 contains each input parameter and the name of the corresponding subroutine which must be supplied if the user wishes to create his own function. Table 3.1 also includes the independent variables which may be used to define each parameter. For further details involving user-supplied subroutines, see Appendix C.

If the thermal conductivity of a material is anisotropic, then it must be defined in user-supplied subroutine CONDTN (see section 3.6.7).

3.6 Card Description

A detailed description of the card input is presented below. Except for Card 1, the M cards, and the deck composed of the IT cards, the input data are arranged on each card in 9-column fields. All integers must be right-adjusted, i.e., the last digit of each integer must appear in a column which is a multiple of 9. Except for the IT cards, columns 73 through 80 of each card are reserved for identification to aid the user in the preparation and handling of the data.

3.6.1 Card 1 - Title of Problem

This card, which can contain alphanumeric characters in the first 72 columns, contains a descriptive title for the problem. The card itself cannot be omitted although it may be left blank.

3.6.2 Card 2 - Input Parameters

All eight entries in this card are integers.

a Maximum CPU Time

When one submits a job to the computer, the maximum CPU time that the job is expected to run is indicated on the CLASS card. If the CPU time exceeds this time, then the job will be pulled by the system without printing out the current temperature distribution. In order to prevent this, the maximum CPU time (seconds) for the IBM 360/91 at ORNL or IBM 360/195 at ORGDP is specified as the first entry in Card 2.

As a safety factor the code subtracts five seconds from this specified time. If the code determines that the job is being run on the IBM 360/75, then it multiplies the number in this entry by 5. After each iteration or time step, a check is made to see if the CPU time exceeds the specified time. If so, it completes all of the output options which are specified and attempts to read the data for the next problem, if any. This time should be less than the CPU time specified on the CLASS card.

b. *Type Geometry*

The HEXEREI2 Program at present offers two possible geometries which are basically cylindrical. These are listed below.

- 1) R- θ -Z - Hexagonal mesh surrounded by cylindrical mesh.
- 2) R- θ -Z - Hexagonal mesh only.

c. *Total Number of Regions*

The total number of regions of the entire configuration is entered as the third entry. A maximum of 100 regions is allowed.

d. *Total Number of Materials*

The fourth entry of this card contains the total number of different materials. There can be a maximum of 50 materials.

e. *Total Number of Heat Generation Functions*

The total number of different heat generation functions (a maximum of 20) is the sixth entry of Card 2. This entry is left blank if there are no heat generation functions.

f. *Total Number of Initial Temperature Functions*

This entry (the seventh on Card 2) is the total number of different initial temperature functions, up to a maximum of 25. If there are no initial temperature functions, this entry will be left blank, and the program will assume that the initial temperature distribution is zero degrees.

g. *Total Number of Boundaries*

The eighth entry of Card 2 is the total number (maximum of 50) of boundary conditions. If there are no boundary conditions explicitly specified in the input data, this entry is left blank.

3.6.3. Card 3 - Input Parameters

Each entry on this card is an integer.

a. *Gross Lattice Size*

The first three entries of this card contain the total number of gross lattice lines in (1) R direction, (2) θ direction, and (3) Z direction. There can be a maximum of 50 gross lattice lines along each axis.

b. *Total Number of Analytical Functions*

The fourth entry is the total number of different analytical functions. There can be a maximum of 25 of these functions. If there are no analytical functions, then this entry will be left blank.

c. *Total Number of Tabular Functions*

The total number of tabular functions is the fifth entry of this card. There can be a maximum of 25 functions. If there are no tabular functions, then this entry is left blank.

d. *Temperature Units*

For problems involving radiation, the temperature units *must* be either °F or °C. Entry 6 is used only for problems involving radiation. It indicates that the units of temperature are either in degrees Fahrenheit or degrees Celsius.

<u>Entry 6</u>	<u>Temperature</u>
0 or Blank	°F
1	°C

When the coolant calculations are used (section 3.6.16), then the temperature units must be in °F. This restriction will be removed in future code improvements.

e. *Type of Output Option*

The seventh entry on this card is a flag used to control the amount of output for each printout. The options are as follows:

<u>Entry 7</u>	<u>Option</u>
1	Output three-dimensional temperature distribution for both solid and coolant, coolant exit temperatures, and summary information.
2	Output is the same as 1 except coolant exit temperatures are omitted.
3	Output three-dimensional solid temperature distribution and the summary information.
4	Output three – dimensional coolant temperature distribution and the summary information.
5	Output summary information only. (Default)

f. *HTGR Input Unit Number*

The eighth entry on this card specifies the number of the unit number from which the HTGR data will be read. If left blank, the data will not be read.

3.6.4. Card 4 - Input Parameters

Each entry on this card is an integer.

a. *Transient Output*

For transient problems, the output may be specified in either of two ways:

- (1) The temperature distribution may be printed out at equally spaced times. To specify this option, the first entry on Card 4 must contain the number of initial time steps between outputs, and the second entry must be blank. For example, if a value of 5 is entered, the temperature distribution will be printed at times where spacing is equal to five times the initial time step.
- (2) The temperature distribution may be printed out at unequal time increments. To choose this option, the first entry on Card 4 may be left blank and the second entry must contain the number of times the temperature distribution will be printed out. The actual output times will be entered on the O cards.

The temperature distribution may be printed out 100 times for each transient portion of the problem. For steady-state-only calculations, the first and second entries may be left blank. The temperature distribution is automatically printed out prior to the first time step for transient calculations and prior to the first iteration for steady-state problems.

b. *Graphical Output*

The third entry on Card 4 indicates whether or not graphical output is desired. At times it is desirable to have plots of the temperature distribution to aid in the interpretation of the results from HEXEREI2. By entering a nonzero integer in this field, a data set will be created which contains the temperature distributions along with certain parameters which identify the problem. The absolute value of the integer identifies the unit number on which the data set is to be created. If it is a positive integer, a new data set is created. If it is negative, then the temperatures from this run will be added to those from a previous run. The data set containing the temperatures from the previous run will be read from the unit whose number is one less than the absolute value of this entry (or one less than the unit number of the data set to be created). This data set can then be used by a plotting package to create various plots to include temperatures versus time, temperature profiles and isothermal plots. If this option is invoked, the user must supply a DD card to define the data set. The records are variable in length. The longest and most frequently written record is $8(N + NBC)$ bytes long where N is the number of nodes and NBC is the number of boundary conditions. See Appendix A for a sample DD card. The plotting package will be documented in the near future.

c. *Frequency of Output for Plots*

This entry, the fourth on Card 4, specifies the number of time steps between each output of the temperature distribution on the data set defined in section 3.6.4(b). If this entry is blank or zero and if the preceding entry is nonzero, then the temperature distribution will be written on the data set each time that a normal printout as defined in section 3.6.4(a) occurs.

d. *Special Monitoring of Temperatures*

The user may wish to tabulate the temperatures of a few nodes as a function of the number of iterations or time steps. In this way one can keep track of what is happening at a few nodes of interest without getting excessive output by having to

print out the entire temperature distribution. This output option is in addition to the standard output of the temperature distribution. To invoke this option, the number of iterations between printouts for steady-state calculations or the number of time steps between printouts for transient calculations is entered in the fifth field of Card 4. If this entry is nonzero, the node numbers whose temperatures are to be printed out are specified on the S cards. If this option is not desired, this entry is left blank.

e. *Initial Temperature Input Unit*

The sixth entry on this card specifies the unit number from which the explicitly specified lattice-point initial temperatures are read. If the entry is a positive integer, then it specifies the unit number from which the initial temperatures are read in formatted form. The records vary in length with the maximum size being 80 characters. If it is a negative integer, then its absolute value specifies the unit number from which the initial temperatures are read in unformatted form. If there are no initial temperatures explicitly specified, then this entry is left blank. If the unit specified is other than the standard card input, the user must insure that the appropriate DD card has been supplied to describe the data set.

f. *Final Temperature Output Unit*

In addition to the normal output, the user may wish to have the final temperature distribution saved in some manner to facilitate the restarting of the problem. If this option is desired, the unit number on which the final temperature distribution is to be written is entered on the seventh entry on Card 4. If the entry is a positive integer, then it specifies the unit number on which the final temperature distribution is to be written in formatted form. The records vary in length with the maximum size being 80 characters. If the entry is a negative integer, then its absolute value specifies the unit number on which the final temperature distribution is to be written in unformatted form. If the user does not wish to save the final temperature distribution, then this entry is left blank. If this entry is nonzero, the user must see that the appropriate DD card has been supplied to correctly identify the specified unit (see Appendix A).

As an example, if the user feels that he may wish to restart the problem at some future date, he may enter the unit number for the standard card punch (7) as the seventh entry on Card 4. Then, when the user actually restarts the problem, he will enter the standard input unit (5) in entry 6 on Card 4. The punched output from the previous run (the final temperature distribution) will be the IT deck in the current run.

g. *Problem Status Unit for Remote Users*

This feature is designed to allow remote users to determine the status of their problem without having to wait for normal computer turnaround. If a positive number is entered in the eighth field of Card 4, then this designates the unit number on which error messages and selected information concerning the status of the problem is written. The user must supply the appropriate DD card to define the specified unit. As an example, the user may wish to use this option to have information stored on his PDP-10 file. It can then be printed out on a remote terminal. One can locate data errors and resubmit the corrected problem without having to wait for the main listing from the computer. Using this concept, one could possibly speed up turnaround time.

3.6.5. Card 5 - Input Parameters

Each entry on this card is a floating-point number except for entries 1, 2, and 6, which are integers.

a. *Type of Problem*

The first entry on Card 5 specifies the type of problem. It may be steady-state only, transient only, or combinations of steady-state and transient calculations. The number to be entered is in accordance with the following list:

1 S-S only	-1 trans. only
2 S-S, trans.	-2 trans., S-S
3 S-S, trans., S-S	-3 trans., S-S, trans.
4 S-S, trans., S-S, trans.	-4 trans., S-S, trans., S-S
.	.
.	.
.	.
n S-S, trans., S-S, trans. S-S, etc.	-n trans., S-S, trans., S-S trans., etc.

If, for example, a 3 is entered, the program will first perform a steady-state calculation at time zero; next the transient calculation; then a steady-state calculation at the final transient time using the final transient temperatures as an initial estimate for the steady-state temperatures. Any number of combinations is allowable; however, machine running time should be considered. (Note, careful consideration must be given to formation of time-dependent and heat generation functions with a problem type which uses more than one transient and one steady-state pass.)

b. *Maximum Number of Steady-State Iterations Allowed*

If the maximum number of steady-state iterations is reached, and the convergence criterion is not satisfied, then the program will write "END STEADY-STATE CALCULATIONS, CONVERGENCE NOT SATISFIED" on the printer and will terminate the calculation and call for the next problem. Normally, 200 to 500 iterations are sufficient to converge to the solution. In the event that the convergence criterion is not satisfied, one may wish to save the final temperature distribution in order to restart the problem as discussed in section 3.6.4(f). This entry is the second of Card 5 and is left blank for pure transient problems. If it is zero or blank for steady-state problems, then a default of 500 is used.

c. *Steady-State Convergence Criterion*

This entry, which contains the value of ϵ in Eq. (2.3.2.7), affects the steady-state type of calculation and may be left blank for a transient-only problem. The steady-state calculation will continue until the convergence criterion is met. Since the criterion which insures convergence varies from case to case, the user must rely on his own judgment and experience in determining the correct value for his particular problem. If it is left blank for steady-state problems, the code assumes $\epsilon = 10^{-5}$. This is the third entry of Card 5.

d *Steady-State Over-Relaxation Factor*

The value of β (see Eq (2 3 2 6)) is the fourth entry on this card and its value must be in the range $1 \leq \beta < 2$. This entry may be left blank for transient-only problems. If it is left blank for steady-state problems, the code assumes that the initial value of β is 1.9.

e *Time Increment*

This entry (the fifth on Card 5) contains the initial time increment, Δt (see Eq (2 3 3 1)), for transient problems that will be solved by one of the explicit techniques. For transient problems which will be solved using the implicit procedure, this entry must be left blank. For the Classical Explicit Procedure, the time increment must satisfy certain criteria to insure stability (see Eq (2 3 3 2)). The HEXEREI2 Program calculates the stability criterion and writes, "THE STABILITY CRITERION IS (the value of criterion) FOR POINT (lattice point yielding the above value)". If the input time increment is less than or equal to the stability criterion, then the code writes, "THE INPUT TIME INCREMENT SATISFIES THE STABILITY CRITERION," and uses the input value. If the input time increment is too large (but not more than a factor of 10 too large), the program sets Δt equal to the calculated stability criterion and writes, "THE INPUT TIME INCREMENT DOES NOT SATISFY THE STABILITY CRITERION—THE TIME INCREMENT IS NOW = (value of Δt used in subsequent calculations)". If the input Δt is too large by more than a factor of 10, the program writes, "CASE DELETED—INPUT TIME INCREMENT EXCEEDS THE STABILITY CRITERION BY MORE THAN A FACTOR OF 10," and goes on to the next problem. For steady-state-only problems, this entry may be left blank.

f *Levy's Explicit Method Option*

The sixth entry (an integer) is the factor by which the stable time increment is multiplied to form the time increment for Levy's explicit method. If the entry is blank or less than 2, then Levy's method will not be used. If Levy's explicit method is specified, then the problem is not terminated if the input time increment is more than an order of magnitude larger than the stability criterion as noted in section 3.6.5(e). Instead, the time increment is initially set equal to the stability criterion. HEXEREI2 always executes 10 time increments using a stable time step prior to using Levy's explicit method. If the entry is equal to 1, the time step size equal to the stability criterion will be used at all times. The input time step size must not be larger by a factor of 10 than the stability criterion for the problem to start the transient. This entry may be left blank for type 1 problems.

g *Initial Time*

This entry affects both transient and steady-state problems. It indicates the initial time for problems with a negative type and the time at which the time-dependent functions are evaluated for problems whose types are greater than zero. For example, assume that a type 4 problem is run. Let t_o represent the initial time (entry 7) and t_f represent the final time (entry 8). Then, the time-dependent functions will be evaluated at t_o for the first steady-state calculation. The first transient calculation will begin at time t_o , using the results of the steady-state calculations as the initial temperatures, and will stop at t_f . The second steady-state calculation will then be performed, and the time-dependent functions will be evaluated at t_f . The results from the transient calculation will be used as the first approximation to the steady-state temperature distribution. Finally, the last transient calculation will begin at t_f and continue until $t_f + (t_f - t_o)$. The value $(t_f - t_o)$ will be added to the output times from the first transient calculation to obtain the output times for the second transient case. If entry 7 is left blank, then t_o will be zero.

(2) Heat Generation of Region

This entry contains the number (an integer) of the heat generation function associated with the region given on the first card of this pair. If this entry is left blank, then the code assumes that the region does not generate heat. This entry is left blank for a gap region.

(3) Boundary Numbers

The remaining entries of this card are the boundary numbers defining the boundary conditions corresponding to the six boundaries of the region described by the first card of this pair. These numbers are integers and are entered as follows:

Each entry contains the boundary number of the region boundary appearing in the corresponding entry of Card R1. A boundary condition cannot be specified on a boundary dividing two regions. The entry is left blank for boundaries which are insulated.

3.6.7. *Material Data (Card M)*

An M card and possibly a PC card are required to describe each material. The total number of M cards is the fourth entry on Card 2. There can be up to 50 materials in a HEXEREI2 analysis. If the thermal conductivity of a material is anisotropic, then it is specified as being temperature-dependent, and the associated temperature-dependent function is specified as being user-supplied. The user then programs the anisotropic algorithm for that material in subroutine CONDTN. The following labeled common must be added to subroutine CONDTN:

COMMON /THBSBC/ NBDTP, NDIR

The variable NDIR will contain a value of 1, 2, or 3 indicating the thermal conductivity is to be evaluated along the R, θ , or Z axis, respectively. Both variables NBDTP and NDIR are INTEGER*4.

(1) Material Number

The first entry, an integer, contains the number of the material which is to be described. Materials are numbered consecutively (each different material has a number) beginning with number 1 up to a maximum of 50 materials.

(2) Material Name

The second entry, which must begin in Column 11 and may extend through Column 18, contains the name of the material. This name, which may consist of up to eight alphanumeric characters, is used to aid in identification of output data.

(3) Constant Thermal Properties

Entries 3, 4, and 5 are floating-point numbers and respectively contain the constant thermal conductivity, constant density, and constant specific heat of the material. These entries correspond to the factor, P_0 , in Eq. (3.4). Since the density and specific heat are not used in steady-state calculations, entries 4 and 5 may be left blank for type 1 problems.

(4) Temperature-Dependent Thermal Properties

Entries 6, 7, and 8 (integers) identify the analytical or tabular functions describing the thermal conductivity, density, and specific heat, respectively, as a function of temperature. These entries correspond to subscript m in Eq. (3.5). Entries 7 and 8 may be left blank for type 1 problems.

h. *Final Time for Transient Calculation*

The final time for the first transient calculation is specified as the eighth entry on Card 5. For steady-state problems, this entry may be left blank.

3.6.6. *Region Data*

Each region is described by two cards which must appear in pairs. The cards are repeated for each region. The number of pairs of cards is the third entry on Card 2. There must be at least one region for each problem.

a. *Card R1*

(1) Region Number

This entry contains the number of the region to be described. Regions are to be numbered consecutively beginning with number 1 up to a maximum of 100 regions. Refer to section 3.2 for further details about regions. This entry is the first on Card R1 and must be an integer.

(2) Material in Region

The second entry of this card indicates by an integer the number of the material which occupies the region named in the first entry of this card. This entry is left blank if the region does not contain a material (gap region).

(3) Region Dimensions

Dimensions of the region boundaries are entered as floating-point numbers and are arranged in the following order:

- a) smaller dimension of R region boundary
- b) larger dimension of R region boundary
- c) smaller dimension of θ region boundary (not used at present)
- d) larger dimension of θ region boundary (not used at present)
- e) smaller dimension of Z region boundary
- f) larger dimension of Z region boundary

These dimensions are entered as the third through eighth entries, respectively. The region dimensions must be nonnegative. In R- θ -Z geometry, R = 0.0 must be located at the origin with R increasing radially; Z is not necessarily begun at the origin, but must extend upward; and θ lines or planes increase clockwise to produce a right-hand system. All dimensions are entered as their distance from the origin.

b. *R2 Card*

This card must be included in conjunction with the appropriate R1 card, even if it is blank.

(1) Initial Temperature of Region

The initial temperature function number of the region specified by the first entry of Card R1 is entered as an integer. If this entry is left blank, then the program assumes that the initial temperature for the region is zero. This entry is left blank for a gap region.

- (3) The third entry, a floating-point number, contains the boundary temperature, T_b (corresponds to P_0 in Eq. (3.4)).
- (4) Since the boundary temperature can be a function of time, the fourth entry on this card contains the time-dependent parameter (an integer) which corresponds to subscript l in Eq. (3.5). This parameter identifies an analytical or tabular function. If the boundary temperature is independent of time, then this entry will be left blank.

b. *Card B2*

Each entry on Card B2 is a floating-point number except entry 6 which is an integer. This card is left blank for a type 2 boundary condition.

- (1) Entry 1 contains the heat transfer coefficient, h_c , for forced convection. (Corresponds to P_0 in Eq. (3.4).)
- (2) Entry 2 contains the coefficient h_r , for radiation. (Corresponds to P_0 in Eq. (3.4).)
- (3) Entry 3 contains the coefficient, h_n , for natural convection. (Corresponds to P_0 in Eq. (3.4).)
- (4) The fourth entry contains the exponent, h_e , for natural convection (or other nonlinear heat transfer process). (Corresponds to P_0 in Eq. (3.4).)
- (5) The fifth entry contains the prescribed heat flux, h_f , across the boundary. (Corresponds to P_0 in Eq. (3.4).)
- (6) The time- and temperature-dependent parameter flag, an integer, is the sixth and final entry on the B2 card. If any of the five preceding parameters are functions of time or temperature, then additional information must be entered on B3 and/or B4 cards. The time- and temperature-dependent flag indicates whether or not the B3 and B4 cards are present for this particular boundary condition. Its value is determined according to the following table:

<u>Entry Six</u>	<u>Additional Cards</u>
0	None
1	B3 Only
2	B4 Only
3	B3 and B4

c. *Card B3*

All five entries on Card B3 are integers. Each integer identifies the analytical or tabular function that defines the time-dependent function associated with the respective parameter on Card B2. Each entry corresponds to subscript l in Eq. (3.5). If an entry is zero, then the associated parameter is not time-dependent.

3.6.8. Heat Generation Function Data (Card G)

Each different heat generation function is numbered, beginning with number 1, consecutively up to a maximum of 20 such functions. As indicated in Table 3.1, the heat generation function associated with a region may be dependent on position, time and temperature. Thus, the heat generation function data cards indicate the function number, the volumetric heat generation rate (corresponds to P_0 in Eq. (3.4)), the time-dependent function parameter, the temperature-dependent function parameter, the R-dependent function parameter, the θ -dependent function parameter, and the Z-dependent function parameter, arranged in that order, respectively. The parameters in entries 3 through 7 refer to analytical or tabular functions defining l , m , i , j , and k , respectively, in Eq. (3.5). If the heat generation rate per unit volume does not vary along an axis or if the problem is one- or two-dimensional, then the position-dependent function parameter corresponding to that coordinate will be left blank, and the associated function value will be set equal to 1.0. If the heat generation rate per unit volume does not vary with time or temperature, then the time- or temperature-dependent function parameter will be omitted and the associated function value or values will be set to 1.0. The first entry on this card is an integer, the second is a floating-point number and the remaining five are integers. The total number of cards is indicated by the sixth entry on Card 2. The heat generation rate for a region may be positive (heat source) or negative (heat sink). The G cards are omitted if entry 6 on Card 2 is blank or zero.

3.6.9. Initial Temperature Function Data (Card I)

Each different initial temperature function is given a number. Beginning with number 1, the initial temperature functions are numbered consecutively up to a maximum of 25. Since the initial temperature function associated with a region can be a function of position, then this data card consists of five entries. All are integers except the second which is a floating point number. The first entry contains the initial temperature function number. The second entry contains the constant factor describing the initial temperature function. This term corresponds to P_0 in Eq. (3.4). The remaining entries, which identify analytical or tabular functions, contain the R-, θ -, and Z-dependent function parameters corresponding to the subscripts i , j , and k , respectively, in Eq. (3.5). If the initial temperature does not vary along a particular axis of the region, then the position-dependent function parameter associated with the coordinate will be left blank, and the corresponding function value will be set equal to 1.0. The total number of cards is the seventh entry on Card 2, and if this entry is blank or zero, then the I cards are omitted and the initial temperature distribution is assumed to be zero.

3.6.10. Boundary Data (Cards B1, B2, B3, and B4)

Excluding insulated or contact-type boundaries, each unique boundary is numbered consecutively up to a maximum of 50. The B1 and B2 cards must be omitted if the eighth entry on Card 2 (this entry indicates the total number of boundary conditions specified) is blank or zero.

a. Card B1

- (1) The first entry on Card B1 is an integer and contains the boundary number.
- (2) The second entry (an integer) indicates the type of boundary. HEXEREI2 offers two boundary types which are numbered corresponding to the following:
 - 1 implies surface-to-boundary, and
 - 2 implies prescribed surface temperature.

If this entry is blank or zero, then no heat transfer connections will be made and the boundary will be treated as an insulated boundary.

d. *Card B4*

This card is just like Card B3 except each integer identifies the analytical or tabular function that defines the temperature-dependent function associated with the respective parameter on Card B2. Each entry corresponds to subscript m in Eq. (3.5).

In evaluating a particular parameter, the code uses Eq. (3.4) where P_o is the appropriate value from Card B2, $F(t)$ is the time-dependent function as defined by Card B3 and $F_m(T)$ is the temperature-dependent function as defined by Card B4.

The B1 and B2 cards must appear in pairs, and a pair is entered for each boundary. B3 and B4 cards, if any, must follow their respective B1 and B2 cards.

3.6.11. *Lattice Description*

For each axis, gross lattice data are entered on two sets of cards, the first set specifying the lattice dimensions and the second indicating the mesh division between gross lattice lines. All of the numbers on the cards on the first set (L cards) are of the floating-point type and are entered, by specifying all of the gross lattice dimensions in each direction, sequentially on one or more cards. The cards of the second set (N cards) specify the number (an integer ≥ 1) of equal increments which are between the gross lattice lines whose dimensions are given on the cards of the first set. In particular, an entry on an N card specifies the number of equal increments between the gross lattice line in the corresponding entry on the related L card and the gross lattice line immediately following it. This procedure is repeated for each coordinate. The R coordinate is modified slightly since the user does not necessarily know how many nodal columns will be required in a problem. The procedure followed is that the first radial gross grid line input should be zero and the second should be the radius of the circle that will lie between the last two concentric hexagonal rings of the hexagonal mesh. The HTGR active core radius is normally used here and the code insures that this radius will be contained within the hexagonal mesh with a full hexagonal ring surrounding it. The first entry on the N1 card should be a negative one to flag the hexagonal mesh generator. The rest of the L1 and N1 cards are utilized as described above. The L2 and N2 cards currently are read but not utilized. The axial grid generated assumes that the first and last nodal planes consist of half cells, thus these nodal planes coincide with the upper and lower model boundaries.

a. *Card L1*

The L1 cards correspond to the R coordinate, and the number of entries corresponds to entry 1 of Card 3. If there are more than 8 entries, subsequent cards are used. The first entry should be zero, and the second equal to the active core radius if used in HTGR analysis.

b. *Card N1*

The N1 cards correspond to the R coordinate. There will be one less entry here than there are on the L1 cards. Additional cards are used for more than eight entries. The first entry should be -1 if a hexagonal mesh between the first two entries on the L1 card is desired.

c. *Card HXP*

The HXP card applies to the R coordinate only and is input only if hexagonal mesh generation is utilized. The single entry is a floating-point number representing the distance between nodal columns in the hexagonal mesh zone.

d. *Card L2*

The L2 cards correspond to the θ coordinate, and the number of entries corresponds to entry 2 of Card 3. If there are more than eight entries, subsequent cards are used.

e. *Card N2*

The N2 cards correspond to the θ coordinate. There will be one less entry here than there are on the L2 cards. Additional cards are used for more than eight entries.

f. *Card L3*

The L3 cards correspond to the Z coordinate, and the number of entries corresponds to entry 3 of Card 3. If there are more than eight entries, subsequent cards are used.

g. *Card N3*

The N3 cards correspond to the Z coordinate. There will be one less entry here than there are on the L3 cards. Additional cards are used for more than eight entries.

3.6.12. *Analytical Function Data (Cards A1 and A2)*

Each analytical function, defined in Eq. (3.1), is described by an A1 card and one or more A2 cards.

a. *Card A1*

Each different analytical function is numbered, and there can be a maximum of 25 such functions. The first entry on Card A1 is the unique analytical function number, an integer. The second entry, also an integer, is the number of coefficients, A_i , which are on the A2 cards. If this entry is blank or zero, then the code assumes that a user-supplied function will be supplied for the parameter which uses this particular analytical function.

b. *Card A2*

The A2 cards contain from one to four ordered pairs, where each ordered pair is defined as follows: the first element of an ordered pair consists of an integer i ; the second element consists of the value of the coefficient A_i . The A2 cards will be continued until each coefficient in the analytical function is defined. If the second entry on Card A1 is blank or zero, then the related A2 card is omitted.

As an example, the cards which are necessary to describe $f(x) = 100\sin 5x + \exp(-3x)$ are presented in Table 3.2

The total number of analytical functions is the fourth entry on Card 3. The A cards are omitted if this entry is blank or zero.

Table 3.2. A Cards Necessary to Describe the Function
 $100\sin 5x + \exp(-3x)$.

Column	9	18	27	36	45	54	63	72	75
	1	4							A1
	8	100.0	9	5.0	6	1.0	7	-3.0	A2

3.6.13. Tabular Function Data (Cards T1 and T2)

Each tabular function is numbered consecutively beginning with one up to a maximum of 25 functions. The tabular function is assumed to be a set of linearly connected points. Each ordered pair contains an independent variable and its functional value; viz., $t_1, G(t_1)$; $t_2, G(t_2)$; etc. A maximum of 25 points (pairs) is allowed, and linear interpolation is performed between the points by the program. The values of t must be entered in ascending order.

a. Card T1

The first entry on the T1 card (an integer) is the tabular function number. The number of points (an integer) is the second entry.

b. Card T2

The T2 card contains the first four ordered pairs, all floating-point numbers. If there are more than four ordered pairs in the function, they are entered on subsequent T2 cards.

Figure 3.6 is an example of an acceptable tabular function. The input for this example is presented in Table 3.3.

Fig. 3.6. Plot of a Sample Tabular Function

Table 3.3. T Cards Necessary to Describe the Sample Tabular Function Depicted in Fig. 3.6.

Column	9	18	27	36	45	54	63	72	Card
	1	9							T1
	0.0	25.0	1.0	30.0	2.0	30.0	2.5	35.0	T2
	3.5	35.0	4.0	30.0	4.5	30.0	5.5	20.0	T3
	5.501	10.0							T4

The total number of tabular functions is the fifth entry of Card 3. The T cards are omitted if this entry is blank or zero.

3.6.14. *Output Times (Card O)*

Each entry on this card is a floating-point number. Since the second entry on Card 4 indicates the total number of output times which are to be read, the O cards are omitted if this entry is blank or zero. The transient output times are entered in chronological order on the O cards. There can be eight entries per card, and the O card is repeated as often as necessary to describe the output array. The maximum number of output times is 100, not counting the automatic printout which occurs prior to the initial time step.

3.6.15. *Node Numbers for Special Monitoring of Temperatures (Card S)*

As optional output, one may specify up to 20 nodes whose temperatures will be printed out as a function of the number of iterations for steady-state calculations or the number of time steps for transient calculations. The first entry on Card S contains the total number of nodes whose temperatures are to be tabulated. The remaining fields contain the actual node numbers. If more than seven nodes are specified, their numbers will appear on additional cards. All entries on the S cards are integers. The frequency for printing out the temperatures of such nodes appears as the fifth entry on Card 4, and the S card or cards are omitted if this entry is blank or zero.

3.6.16. *HTGR Input Block*

The information contained on the following cards is necessary to modify HEXEREI2's parameters to accurately represent an HTGR. Inclusion of some of the following data is necessary to model a general multichannel flow problem. This block of data cards will be read from the logical unit number input on card 3 (entry 8). It may be read from unit 5 (normal card input) if so specified on card 3 and placed in the relative position in the data stream as shown in section 3.8.

The first cards in this block are general cards required whenever any of the HTGR input block is required.

a. *Card HTGR*

All entries but the first, which is an integer, are floating-point numbers.

- 1) Entry 1 contains the number of REG cards to follow.
- 2) Entry 2 contains the coolant pressure in the upper plenum (corresponds to P_o in Eq. (3.4)).
- 3) Entry 3 contains the coolant temperature in the upper plenum (used in analysis without reverse flow option, section 2.3.7, and corresponds to P_o in Eq. (3.4).)
- 4) Entry 4 contains the coolant molecular weight (corresponds to P_o in Eq. (3.4)).
- 5) Entry 5 contains the coolant mass flow rate into the upper plenum (corresponds to P_o in Eq. (3.4)).
- 6) Entry 6 contains the diameter of the larger coolant channels.
- 7) Entry 7 contains the diameter of the smaller coolant channels.
- 8) Entry 8 contains the convergence criterion for the coolant calculations (see sections 2.3.6 and 2.3.7).

b. *Card HTFN*

All entries are integers for this card and entries 1 through 4 contain the time function number for the coolant pressure, temperature, molecular weight and mass flow rate, respectively. The fifth entry contains a number whose absolute value is the unit number for the output of the CACS calculation results and is the flag enabling the CACS calculational package. If it is nonzero, then the ACH card is read and CACS calculations are performed. If it is positive, open loop calculations are to be performed and, if negative, closed loop calculations are to be performed.

c. *Card REG*

This group of cards replaces initial information in the model parameters for the nodal columns referenced. Initial values are zero. The first three entries are integers while the last four are floating-point numbers.

- 1) Entry 1 contains an integer which is the nodal column number (NH) to which the data on this card pertains.
- 2) Entry 2 contains the integer number of small coolant channels within the region associated with the nodal column above.
- 3) Entry 3 contains the integer number of large coolant channels within the region.
- 4) Entry 4 contains Region Power Peaking Factor (RPPF).
- 5) Entry 5 contains the fraction of the region that contains active fuel elements (PCACFE). The heat generated within a node is then calculated as

$$P = P(t,r,\theta,z) \text{ RPPF(NH) PCACFE(NH)}$$

- 6) Entry 6 contains the orifice area setting for HTGR transients.
- 7) Entry 7 contains the input coolant mass flux.

d. *Card CSB*

This card describes the two expansions in the core support blocks and lower reflector elements as described in sections 2.3.6 and 2.3.7. All entries are floating-point and the first two contain the diameters for the first and second expansions. The next two entries contain the z coordinate of the two expansions.

e. *Card CAT*

All entries are floating-point and contain the orifice maximum flow area (also the orifice cylinder area); the orifice minimum flow area; the orifice cylinder length and hydraulic diameter; the flow areas of the upper reflector plenum, first and second expansions in the lower reflector/core support blocks. See section 2.3.7 for a description of these parameters.

f. *Card ACH*

The model parameters that describe the CACS system are contained on this card (see section 2.3.10). All entries are floating-point and contain the circulator design pressure drop, pressure, inlet temperature, mass flow rate, and speed in the first five entries. The sixth entry contains the limit on circulator motor torque. The seventh contains the limit on cooling water maximum temperature. The last entry contains the convergence criterion for the closed loop analysis.

3.6.17. *Initial Temperatures*

These cards (or card images) are generated as output by HEXEREI2 if a positive number appears in entry 7 on Card 4 and, generally, they are used only when restarting a job by merely inserting the generated deck at this location in the original deck and resubmitting the job. As noted in section 3.6.4(e), one must enter the unit number of the card reader in the sixth field of Card 4. Entry 6 on Card 4 specifies the unit on which these data are to be read. If the unit specified is other than the standard card input, the user must insure that the appropriate DD card has been supplied to describe the unit. If entry 7 on Card 4 is nonzero, then the code generates these data at the end of a problem and writes it on the unit specified in entry 7. However, the user must insure that the appropriate DD card has been supplied to correctly identify the unit specified on entry 7. If entry 6 on Card 4 is blank or zero, then these cards are omitted.

Since the user may wish to explicitly specify the initial temperature at some point or points, a description of these data is given below.

a. *Job Description (Card IT1)*

This card image gives a descriptive title and can contain alphanumeric characters in the first 72 columns. The card may be blank but it cannot be omitted.

b. *Initial Time and Lattice Point Numbers (Card IT2)*

The first entry, a floating-point number which occupies the first ten columns on the IT2 card, specifies the initial problem time. This value overrides the initial time which appears on entry 7 of Card 5. Normally, this deck will have been generated by the code on a previous run. Thus, the code supplies this value as the time for which the following temperature distribution occurs. If the user generates the IT deck, then he must insure that the initial problem time is entered here. The second entry on the IT2 card is an integer and contains the total number of lattice points whose initial temperatures are explicitly specified. It occupies columns 11 through 15 of Card IT2.

c. *Lattice Point Temperatures (Card IT3)*

The IT3 card can contain up to five pairs of numbers with each pair defined as follows:

- (1) The first member of the pair is a lattice point number. It is an integer and occupies a five-column field.
- (2) The second member is the initial temperature of the lattice point whose number appears in the first member. It is a floating-point number and occupies a ten-column field.

The number of pairs to be entered on the IT2 cards is specified on entry 2 of Card IT2. Card IT3 is repeated until all pairs have been described. The temperatures specified by this input data override the corresponding temperatures generated by the I cards.

3.6.18. *Implicit Transient Technique Parameters*

If this problem involves transient calculations (entry 1 on Card 5 is not equal to 1) and if the implicit technique is to be used to calculate estimates to the transient temperature distribution, then the fifth entry (time step) of Card 5 must be left blank and additional data must be supplied on the IP and TP cards. If it is anticipated that similar problems will be run a large number of times such as ones arising in a parametric study, then it is recommended that the variables on the IP and TP cards be optimized since it could significantly reduce the overall computer time.

a. *Card IP*

This entire card or any of its entries may be left blank, and the default values will be used. They are based on experiences with a few two-dimensional RZ models. They are certainly not the best values that can be used for a given problem, but they are probably good starting points. There are eight entries on this card. The first five are floating-point numbers, and the last three are integers.

- (1) The first entry contains one of two convergence criteria which must be met in order for the iterative technique to terminate successfully at each time step. This convergence criterion corresponds to ϵ_1 in Eq. (2.3.3.19). The default is 10^{-3} .
- (2) The second entry contains the second convergence criterion which corresponds to ϵ_2 in Eq. (2.3.3.20). The default for this parameter is 10^{-5} meaning the L_1 norm of the residual must decrease by five orders of magnitude. It has been found that for problems with an abnormally small time step (one approximately equal to the stability criterion or less) this criterion with $\epsilon_2 = 10^{-5}$ may not be satisfied. Increase the time step size, if possible. Otherwise, experiment with the value of ϵ_2 .
- (3) The third entry contains the convergence criterion for problems involving temperature-dependent parameters. This convergence criterion which corresponds to ϵ_3 in Eq. (2.3.3.23) is used in addition to the first two. The default is 10^{-3} .
- (4) This entry, the fourth, defines the implicit technique which will be used to solve the transient problem. It refers to Θ in Eq. (2.3.3.16) and must be chosen so that $0.5 \leq \Theta \leq 1.0$. The default is 0.5.
- (5) The fifth entry defines the initial value of the point successive overrelaxation iteration acceleration parameter (ω in Eq. (2.3.3.16)). It also defines the method that will be used to update the acceleration parameter. If this entry is positive,

then the acceleration parameter will remain constant throughout the calculations and will be equal to the value of this entry. If it is blank or zero, then the acceleration parameter will be optimized empirically as a function of time. This appears to be the best option for nonlinear problems. If it is negative, then the acceleration parameter will be calculated using Carré's technique [Ref. 12]. The absolute value of this entry must be less than 2.0.

- (6) This entry, an integer, defines the number of time steps between attempts to optimize the acceleration parameter empirically (referred to as N_ω in section 2.3.3.3(d)). It is only used when entry 5 is zero or blank. The default value is 1.
- (7) For the case when the acceleration parameter will be updated empirically (entry 5 is blank or zero), then this entry, the seventh, defines the change-in-number-of-iterations criterion (referred to as I_ω in section 2.3.3.3(d)) which must be met before the acceleration parameter will be updated. The default is 5. For the case when the SOR acceleration parameter will be updated using Carré's technique, then this entry defines the number of iterations between updates. The default is 12.
- (8) The last entry is the change-in-number-of-iterations criterion (referred to as J_ω in section 2.3.3.3(d)) which is used to determine when a good estimate to the optimum acceleration parameter has been found. This entry is only used when the acceleration parameter will be updated empirically (entry 5 on this card is blank or zero). The default is 2.

b. *Card TP*

When an implicit scheme is used to solve a transient problem, the time step may be variable. This allows the time step to increase as the solution smooths out and to decrease when some parameter varies rapidly with time. The information controlling the value of the time step is contained on one or more of the TP cards. The size of the time step is automatically adjusted in order to get printouts of the temperature distribution at the specified time. If the size of the coefficients in the system of equations varies by orders of magnitude (10^5 or greater), it has been observed that point-successive overrelaxation iteration may converge very slowly (it may appear to not converge at all). This occurs when the grid spacing or thermal properties vary by orders of magnitude over the problem. It can be observed by examining the stability criterion table in the output. If this appears to be happening, either further subdivide some of the larger nodes or combine some of the smaller ones. In some cases, it may help to use a larger time step size. All seven entries are floating point numbers.

- (1) The first entry is the initial time step.
- (2) After the temperature distribution has been calculated, the current time step is multiplied by a factor. The value of this factor is entered in the second field of the TP card. The default value is 1.0. For many problems whose parameters vary mildly with time and/or temperature, values between 1.0 and 1.3 have been acceptable.
- (3) The maximum value of the time step is the third entry. Once the time step reaches this value, it is no longer increased. The default is 10^{50} .
- (4) The fourth entry contains the maximum time that the time step information on this card applies. If the time reaches this value, then a new TP card is read. The default is 10^{50} .

- (5) The fifth entry contains the maximum temperature change allowed at node over a time step. The time step size is adjusted according to the procedure outlined in section 2.3.3.4. If this entry is blank or zero, then this feature is not invoked in calculating the time step size.
- (6) The sixth entry contains the maximum percent of relative temperature change allowed at a node from one time level to the next one. The time step size is adjusted according to the procedure outlined in section 2.3.3.4. If this entry is blank or zero, then this feature is not invoked in calculating the time step size.
- (7) The seventh and final entry on this card contains the minimum value of the time step. Once the time step size reaches this value, it is no longer decreased. The default is one-tenth of the initial step size.

3.6.19 Blank Card

If the user wishes to solve several problems with one run, he merely inserts a blank card between each problem deck.

3.7 Problem Size Limits

The problem size limits are summarized in Table 3.4.

3.8 Summary and Format of Input

Table 3.5 is a concentrated summary of the format and information needed to prepare the input data deck. Except for the IT cards, columns 73 through 80 of each card are reserved for identification, and the user may or may not choose to punch this information. The table in this column gives proposed identification names for the cards and, in parentheses, the reference sections in the report where more information can be found. The first line in each format box includes the variable name actually used in the program and, in parentheses, the format of that variable. The rest of the box includes a short explanation of the input in that box. In some cards, where free space is available, e.g., G or I cards, additional notes have been included to describe the way the input information will be used in the program itself.

3.9 Output Description

3.9.1 General

The HEXEREI2 program automatically lists the input data and prints out the initial temperature distribution for any type of problem. The final temperature distribution is printed for steady-state problems, and the temperatures are presented as a function of time for transient calculations. As indicated in section 3.6.4(f), the final temperature distribution can be listed on a specified unit (e.g., the card punch). Information can also be written on a specified unit and used by plotting package to create various types of plots (see section 3.6.4(b)).

3.9.2 Input Return

All of the input data are recapitulated in the output, appearing before the temperature distributions. The format is best illustrated by example, and the reader is referred to the sample output (Fig. 3.9.2.1). A table containing each node and its corresponding stability criterion is presented for transient problems.

Table 3.4. Problem Size Limits

Item	Maximum Number
Analytical Functions	25
Boundary Conditions	50
Fine Lattice Lines Along Any Axis	100
Gross Lattice Lines Along Any Axis	50
Heat Generation Functions	20
Initial Temperature Functions	25
Lattice Points	(Variable)
Materials	50
Number of Nodes in Special Temperature Table	20
Number of Nodal Columns	121
Number of Nodal Planes when Using Coolant Calculations	25
Points Per Tabular Function	25
Printout Times	100
Regions	100
Tabular Functions	25

Table 3.5. Summary of Input

Columns 1-9	Columns 10-18	Columns 19-27	Columns 28-36	Columns 37-45	Columns 46-54	Columns 55-63	Columns 64-72	Columns 73-80
JOBDES(1), (18A) Job Description								Card 1 (3 6 1)
MCPU (19) Maximum CPU time for IBM 360/91 or IBM 360/195 in seconds	NGROM (19) 1-R2 with Hexagonal mesh 2-R2 without Hexagonal mesh	NOREGT (19) Total number of regions (Cards R1 & R2) Maximum 100	MATLT (19) Total number of materials (Cards M) Maximum 50	NGENFN (19) Total number of heat generation functions (Cards G) Maximum 20	MINTPN (19) Total number of initial temperature functions (Cards I) Maximum 25	NBUTPT (19) Total number of boundary conditions (Cards B1 & B2) Maximum 50		Card 2 (3 6 2)
IOT (19) Total number of gross lattice lines in 8 direction (Entries in L1) Maximum 50	JOT (19) Total number of gross lattice lines in 8 direction (Entries in L2) Maximum 50	KOT (19) Total number of gross lattice lines in 8 direction (Entries in L3) Maximum 50	NOANAT (19) Total number of analytical functions (Cards A) Maximum 25	NOTBLT (19) Total number of tabular functions (Cards T) Maximum 25	IDEGRE (19) Temperature units If °C, and radiation is involved, enter 1, otherwise, leave blank	JOTYPE (19) Type of output desired 1 - most detailed 3 - least detailed (default)	INTGR (19) Unit for reading HTOR input block If blank or zero HTOR data will not be read	Card 3 (3 6 3)
NDA (19) Number of time steps between printed outputs Next entry must be blank	NPRINT (19) Number of times output will be printed (Entries in 6 card), preceding entry must be blank Maximum 100	ITAPE (19) ITAPE is unit no. for plots ITAPE=0 implies no plots ITAPE < 0 implies plot data will be added to data from previous run read from unit no. ITAPE - 1	NOUPTT (19) Number of time steps between outputs for plots on unit number [ITAPE] If blank, output is at normal printouts	NTS (19) Number of iterations or time steps between output for special monitoring of temperatures (Card S)	JIN (19) Unit number for reading initial temperature distribution Positive formatted Negative unformatted (7 on IBM for cards)	JOUT (19) Unit number for writing out final temperature distribution Positive formatted Negative unformatted (7 on IBM for cards)	IERRO (19) Problem status unit for remote users	Card 4 (3 6 4)
NTYPE (19) Problem type 1 steady state, -1 transient. (See Section 3 6 5(a) for other types)	NOEX (19) Maximum number of steady state iterations (Default 500)	EPT (E9 0) Steady state convergence criterion, -5 (Default, 10 ⁻⁵)	BETA (E9 0) Steady state over-relaxation factor 1.0 < B < 2.0 (Default, 1.9)	DELTA (E9 0) Time increment for transient problem involving explicit technique. Must be left blank for implicit technique	KTDECT (19) Factor by which stable time increment is increased if Levy's explicit method is used. Use 0 or 1 with coolant flow	TIM (E9 0) Initial time (See Section 3 6 5(a) for more information)	FTIME (E9 0) Final time	Card 5 (3 6 5)
NOREG (19) Region number Maximum 100	MATL (19) Region material number (Card M)	SRIN (E9 0) Smaller R region dimension	RAOT (E9 0) Larger R region dimension	TLIT (E9 0) Smaller 8 region dimension	THRT (E9 0) Larger 8 region dimension	ZZBK (E9 0) Smaller Z region dimension	ZZFR (E9 0) Larger Z region dimension	R1 (3 6 6)
ITEM (19) Region initial temperature function number (Card I)	NGEN (19) Region heat generation function number (Card G)	NBDIN (19) Boundary condition on smaller R (Card B)	NBDOT (19) Boundary condition on larger R (Card B)	NBDLT (19) Boundary condition on smaller 8 (Card B)	NBDRT (19) Boundary condition on larger 8 (Card B)	NBDK (19) Boundary condition on smaller Z (Card B)	NBDR (19) Boundary condition on larger Z (Card B)	R2 (3 6 6)
MAT (19) Material number Maximum 50	MATNAM (A2) Material name	CONDUC (D9 0) Conductivity if constant	DENSITY (D9 0) Density if constant. Leave blank for steady state only	SHHEAT (D9 0) Specific heat if constant. Leave blank for steady state only	NGENP (19) Conductivity temperature-dependent function number	NBUTP (19) Density temperature-dependent function number. Leave blank for steady state only	NSHPT (19) Specific heat temperature-dependent function number. Leave blank for steady state only	M (3 6 7)
NGN (19) Heat generation function number Maximum 20	GENO (D9 0) Volumetric heat generation rate if constant	NGNTH (19) Time-dependent function number	NGNTP (19) Temperature dependent function number	NGNR (19) R-dependent function number	NGNTH (19) 8-dependent function number	NGNZ (19) Z-dependent function number	NOTE For the 3rd through 7th entries positive integer implies analytical function and negative integer implies tabular function	G (3 6 8)
INTM (19) Initial temperature function number Maximum 25	TEMPIN (D9 0) Initial temperature, if constant	ITMR (19) R dependent function number	ITMTH (19) 8-dependent function number	ITMTE (19) Z-dependent function number	NOTE For the 3rd, 4th and 5th entries positive integer implies analytical function negative integer implies tabular function			I (3 6 9)
NBUTP (19) Boundary number Maximum 50	NBUTYP (19) Boundary type 1-surface-to-boundary 2-isothermal	BYTEMP (E9 0) Boundary temperature T ₀	NBUTFN (19) Time dependent function number (1) for boundary temperature	Boundary temperature will be established as T ₀ (t)=T ₀ , T ₀ (t) where T ₀ (t) is the 1th analytical function for t > 0 and the (-1)th tabular function for t < 0 NOTE The heat flux will be established as $q = h_f + h_c \Delta T + h_r (T_1^n - T_2^n) + h_n \Delta T ^n$ where $\Delta T = T_1 - T_2$				B1 (3 6 10)
BHCONV (E9 0) Forced convective heat transfer coefficient h _c	BHRAO (E9 0) Radiative coefficient h _r F ₀ or Stefan-Boltzmann constant F Radiation shape factor	BHRAO (E9 0) Radiative coefficient h _r F ₀	BHEXP (E9 0) Natural convective exponent h _n	BFLUX (E9 0) Prescribed heat flux (positive if going to surface) h _f	IMPLUG (19) Parameter flag 0-no additional cards 1-B3 card only 2-B4 card only 3-B5 & B4 cards			B2 (3 6 10)
NBHCTM (19) Forced convective heat transfer coefficient time-dependent function	NBHCTM (19) Radiative coefficient time-dependent function	NBHCTM (19) Natural convective coefficient time-dependent function	NBHCTM (19) Natural convective exponent time-dependent function	NBHCTM (19) Prescribed heat flux time-dependent function	Positive integer implies analytical function Negative integer implies tabular function			B3 (3 6 10)
NBHCTP (19) Forced convective heat transfer coefficient temperature-dependent function	NBHCTP (19) Radiative coefficient temperature-dependent function	NBHCTP (19) Natural convective coefficient temperature-dependent function	NBHCTP (19) Natural convective exponent temperature-dependent function	NBHCTP (19) Prescribed heat flux temperature-dependent function	Positive integer implies analytical function Negative integer implies tabular function			B4 (3 6 10)
RG (E9 0) Smaller 8 gross lattice line dimension	RG (E9 0) Next 8 gross lattice line dimension	RG (E9 0) Next 8 gross lattice line dimension	Can be repeated for up to 30 gross lattice lines (but equal to entry one in Card 3)					L1 (3 6 11)
NDRG (19) Number of DIVISIONS between corresponding 8 gross line and the following line	NDRG (19)	NDRG (19)	Must have one less entry than in L1 cards (maximum of 100 fine lattice lines in each direction) NOTE 1 in any entry will produce a hexagonal mesh that includes the next corresponding 8 gross grid line					N1 (3 6 11)
RFP (E9 0) Nodal column spacing	This card is required only if hexagonal mesh is selected							RXP (3 6 11c)
THG (E9 0) 8	Same as L1 cards except for 8 direction							L2 (3 6 11)
NDTHG (19) 8	Same as N1 cards except for 8 direction							N2 (3 6 11)
ZG (E9 0) Z	Same as L1 cards except for Z direction							L3 (3 6 11)
NDZG (19) Z	Same as N1 cards except for Z direction							N3 (3 6 11)
NANALIT (19) Analytical function number Maximum 10	NAPRM (19) Number of coefficients in A2 cards Maximum 9	NOTE A1 and A2 establish the nth analytical function in the form $F_n(V) = A_{n,1} + A_{n,2}V + A_{n,3}V^2 + A_{n,4} \cos(A_{n,5}V) + A_{n,6} \exp(A_{n,7}V) + A_{n,8} \sin(A_{n,9}V) + A_{n,10} \ln(A_{n,11}V)$						A1 (3 6 12)
NPRM (19) Coefficient index, 1	A(NPRM) (E9 0) Coefficient value, A _{n,1}	NPRM (19) Coefficient index, 1	A(NPRM) (E9 0) Coefficient value, A _{n,1}	NOTE Can be repeated for up to 11 coefficients		NOTE To indicate a user-supplied function, the second entry on Card A1 is left blank and Card A2 is omitted		A2 (3 6 12)
NTABL (19) Tabular function number Maximum 20	NTPRPS (19) Number of pairs in table Maximum 25	NOTE The code uses linear interpolation to obtain the value of this function						T1 (3 6 13)
ARG(1) (E9 0) Independent variable	VAL(1) (E9 0) Corresponding function value	ARG(2) (E9 0) Independent variable	VAL(2) (E9 0) Corresponding function value	NOTE Can be repeated for up to 25 pairs				T2 (3 6 13)
PRTIME (E9 0) First time print-out is desired	PRTIME (E9 0) Second time print-out is desired	PRTIME (E9 0) Third time print-out is desired	Can be repeated for up to 100 printout times (but equal to entry two in Card 4)					0 (3 6 14)
NSN (19) Total number of nodes for special monitoring of temperatures Maximum 20	NDS(1) (19) Node number	NDS(2) (19) Node number	NOTE Node numbers can be repeated for up to a maximum of 20					8 (3 6 15)
NREEN (19) Number of REG cards Maximum 121	PCON (E9 0) Upper plenum pressure (PSIA)	TCOR (E9 0) Upper plenum temperature °F	COOLMW (E9 0) Coolant molecular weight	GTOT (E9 0) Total mass flux (lb/(ft ² hr))	DIAMTR (E9 0) Diameter of large channel (ft)	DIAMIS (E9 0) Diameter of small channel (ft)	EPIC (E9 0) Convergence criteria for coolant calculations	HTOR (3 6 16a)
IACHRN (19) GACS output unit number if ≠ 0, then ACHERONS is used in open loop mode. If > 0 or closed loop mode if < 0	NPTCM (19) Upper plenum pressure time function number	NTCTM (19) Upper plenum temperature time function number	NCHMTM (19) Molecular weight time function number	NGTTH (19) Mass flux time function number	For entries 2 through 5 positive integer implies analytical function negative integer implies tabular function			HTFN (3 6 16b)
NH (19) Refueling region number	NCLCHS (19) Number of small coolant channels	NCLCH (19) Number of large coolant channels	RPPF (E9 0) Region power peaking factor	PCACFE (E9 0) Fraction of region which is active core	ORAREA (E9 0) Office area setting (ft ²)	G (E9 0) Initial mass flux (lb/(ft ² hr))		REC (3 6 16c)
DC (E9 0) Reflector first expansion hydraulic diameter	DE (E9 0) Reflector second expansion hydraulic diameter	ZWEND (E9 0) Second expansion Z coordinate	ZNGOLL (E9 0) Z-coordinate					CSB (3 6 16d)
AREAO (E9 0) Office maximum flow area (ft ²)	OMIN (E9 0) Office minimum flow area (ft ²)	OCLCN (E9 0) Office cylinder length (ft)	DIAGC (E9 0) Office cylinder hydraulic diameter (ft)	AREATP (E9 0) Flow area of upper reflector plenum (ft ²)	AREAE (E9 0) Flow area of first expansion in lower reflector (ft ²)	AREALP (E9 0) Flow area of second expansion in lower reflector (ft ²)		CAT (3 6 16e)
DPO (E9 0) Circulator design pressure rise (PSI)	PO (E9 0) Circulator design pressure (PSIA)	TOCIRC (E9 0) Circulator design temperature (°F)	FRHED (E9 0) Circulator design flow rate (lb/hr)	SPDO (E9 0) Circulator design speed (RPM)	TORMAX (E9 0) Circulator maximum torque (ft-lb)	THOONK (E9 0) Maximum water temperature in CAHE (°F)	EPIC (E9 0) Closed loop convergence criteria	ACH (3 6 16f)
Job Description, Format 18A4								IT1 (3 6 17)
Columns 1-10	Columns 11-15	NOTE INITM specifies number of points entered on the IT3 cards						Columns 76-80
YIM(D10 0) Initial time	INITM (19) Total number of lattice points with explicitly-specified initial temperatures Maximum 1750							IT2 (3 6 17)
Columns 1-5	Columns 6-15	Columns 16-20	Columns 21-30	Columns 31-35	Columns 36-45	Columns 46-50	Columns 51-60	Columns 61-65
R (19) Lattice point number	TI(N) (D10 0) Specified initial temperature of that point	N (19) Lattice point number	TI(N) (D10 0) Specified initial temperature of that point	Can be repeated up to 1750 lattice points				IT3 (3 6 17)
Columns 1-9	Columns 10-18	Columns 19-27	Columns 28-36	Columns 37-45	Columns 46-54	Columns 55-63	Columns 64-72	Columns 73-80
REDOUL (D9 0) Convergence criterion for implicit solution of transient equation. Corresponds to ε ₁ in Eq (2 3 3 19) (Default, 10 ⁻³)	REDUOR (D9 0) Convergence criterion for implicit solution of transient equation. Corresponds to ε ₂ in Eq (2 3 3 20) (Default, 10 ⁻³)	ABSDIF (D9 0) Convergence criterion for implicit solution of transient equation involving temperature-dependent properties. Corresponds to ε ₃ in Eq (2 3 3 23) (Default, 10 ⁻³)	THETA (D9 0) Parameter defining different techniques for transient equation. 0 = THETA = 1.0, 0.5 implies Crank-Nicolson, 1.0 implies Backward Euler, 2.0 implies 2nd order Runge-Kutta, 3.0 implies 4th order Runge-Kutta (Default, 0.5)	BETAT (D9 0) Initial value for BETAT. Corresponds to w in Eq (2 3 3 16). If BETAT = 0, it will be optimized using Carre's technique. If BETAT > 0, it will be constant. 0 < BETAT < 2	NUPETA (19) Number of time steps between attempted acceleration parameter updates. Used in Sec 2 3 3 3(d) (Default, 1)	ITLROO (19) For BETAT = 0, number-of-iterations criterion which initiates acceleration parameter updates. Corresponds to J ₁ in Sec 2 3 3 3(d) (Default, 5)	ITLACI (19) Number-of-iterations criterion to terminate acceleration parameter updates. Corresponds to J ₂ in Sec 2 3 3 3(d) (Default, 2)	IP (3 6 18)
DELTA (D9 0) Initial time step for implicit solution of transient equation	TFACT (D9 0) Factor by which the current time step is multiplied at each time step (Default, 1.0)	TSMAX (D9 0) Maximum time for which time series of time steps of time step applies. When the current time exceeds TSMAX, another TP card is read (Default, 10 ⁵⁰)	TSCHGE (D9 0) Maximum time for which time series of time steps of time step applies. When the current time exceeds TSCHGE, another TP card is read (Default, 10 ⁵⁰)	TPCHGE (D9 0) Maximum time for which time series of time steps of time step applies. When the current time exceeds TPCHGE, another TP card is read (Default, 10 ⁵⁰)	TPCHGE (D9 0) Maximum time for which time series of time steps of time step applies. When the current time exceeds TPCHGE, another TP card is read (Default, 10 ⁵⁰)	TPCHGE (D9 0) Maximum time for which time series of time steps of time step applies. When the current time exceeds TPCHGE, another TP card is read (Default, 10 ⁵⁰)	TPCHGE (D9 0) Maximum time for which time series of time steps of time step applies. When the current time exceeds TPCHGE, another TP card is read (Default, 10 ⁵⁰)	TP (3 6 18)
Blank card if additional problem follows								(3 6 18)
Cards for additional problem if desired

HEXEREI, A MULTI-DIMENSIONAL HEAT CONDUCTION CODE WITH HEXAGONAL MESH, TEMPERATURE-DEPENDENT THERMAL PROPERTIES, NON-LINEAR AND SURFACE-TO-SURFACE BOUNDARY CONDITIONS. THIS CODE IS A MODIFICATION OF THE ONE DESCRIBED IN ORNL-TM-4386(HEATING4), AND HEATINGS DESIGNED TO MODEL THE GENERAL ATOMIC HTGR REACTORS UNDER D3DA OR LOMLFC ACCIDENTS. THE TRANSIENT SOLUTION CAN BE CALCULATED BY AN IMPLICIT TECHNIQUE (CRANK-NICOLSON OR BACKWARDS EULER). IN ADDITION, THIS VERSION IS ESPECIALLY MODIFIED FOR THE THERMAL ANALYSIS OF THE HTGR SAFETY STUDIES TO INCLUDE ONE-DIMENSIONAL COOLANT FLOW IN THE AXIAL DIRECTION FOR THREE-DIMENSIONAL RTZ MODELS WITH HEXAGONAL MESH IN A CYLINDRICAL REGION AROUND THE CENTER OF THE MODEL. HEXEREI WAS WRITTEN BY

W.D. TURNER
G.E. GILES
COMPUTER SCIENCES DIVISION
UNION CARBIDE CORPORATION
OAK RIDGE, TENNESSEE 37830

THIS VERSION OF HEATING CAN HANDLE A MAXIMUM OF 765 LATTICE POINTS.

INPUT RETURN

DATE 10-05-76
TIME 15.24.53

JOB DESCRIPTION-- *****ST.VRIAN D3DA? IMPLICIT 9-3-76*****
THE PROBLEM WILL BE TERMINATED AFTER 330 SECONDS
GEOMETRY TYPE NO. 1
NUMBER OF REGIONS 5
NUMBER OF MATERIALS 2
NUMBER OF HEAT GENERATION FUNCTIONS 1
NUMBER OF INITIAL TEMPERATURE FUNCTIONS 1
NUMBER OF DIFFERENT KINDS OF BOUNDARIES 1
THIS PROBLEM INVOLVES TEMPERATURE-DEPENDENT PROPERTIES.
NUMBER OF POINTS IN GROSS X OR R LATTICE 3
NUMBER OF POINTS IN GROSS Y OR THETA LATTICE 2
NUMBER OF POINTS IN GROSS Z LATTICE 5
NODE PITCH IN THE HEX REGION 3.13333
NUMBER OF NODES PER REFUELING REGION 1
ACTIVE CORE MEAN RADIUS 9.8100
NUMBER OF ANALYTIC FUNCTIONS 2
NUMBER OF TABULAR FUNCTIONS 4
NUMBER OF TRANSIENT PRINTOUTS SPECIFIED 20
SELECTED INFORMATION FOR THIS PROBLEM IS WRITTEN IN UNIT 9
HTGR INPUT WILL BE READ OFF OF UNIT 5
OUTPUT WILL BE OF TYPE 5
PROBLEM TYPE NUMBER 2
CONVERGENCE CRITERION 1.00000000-04
INITIAL BETA 1.90000000
TIME INCREMENT 0.0
INITIAL TIME 0.0
FINAL TIME 5.00000000-01

SUMMARY OF REGION DATA															
NUMBERS AND FCN NUMBER				***** DIMENSIONS *****						----- BOUNDARY NUMBERS -----					
REG.	MATL	INIT	HEAT	LEFT-X-OR	RIGHT-X-OR	LOWER-Y-OR	UPPER-Y-OR	REAR-Z	FRONT-Z	LF-X	RT-X	LO-Y	UP-Y	RR-Z	FT-Z
NO.	NO.	TEMP	GEN.	INNER-R	OUTER-R	LEFT-THETA	RIGHT-THETA			IN-R	OT-R	LF-O	RT-O		
1	1	1	1	0.0000	9.8100	0.0	0.0	3.9000	19.5000	0	0	0	0	0	0
2	2	1	0	9.8100	13.6700	0.0	0.0	1.9000	19.5000	0	1	0	0	0	0
3	2	1	0	0.0	13.6700	0.0	0.0	0.0	3.9000	0	1	0	0	1	0
4	2	1	0	0.0	13.6700	0.0	0.0	19.5000	23.4200	0	1	0	0	1	0
5	2	1	0	0.0	13.6700	0.0	0.0	23.4200	25.3400	0	1	0	0	0	0

***** SUMMARY OF MATERIAL DATA *****

MATERIAL NUMBER	MATERIAL NAME	----- THERMAL PARAMETERS -----		
		-- TEMPERATURE-DEPENDENT FUNCTION NUMBERS --		
		CONDUCTIVITY	DENSITY	SPECIFIC HEAT
1	CURE	1.000000D 01	1.000000D 00	1.000000D 00
		0	0	2
2	REFLECT	1.000000D 01	1.070000D 02	1.000000D 00
		0	0	2

***** SUMMARY OF INITIAL TEMPERATURE DATA *****

NUMBER	INITIAL TEMPERATURE	POSITION-DEPENDENT FUNCTION NUMBERS		
		X OR R	Y OR TH	Z
1	1.000000D 03	0	0	0

***** SUMMARY OF HEAT GENERATION RATE DATA *****

NUMBER	POWER DENSITY	TIME-, TEMPERATURE-, AND POSITION-DEPENDENT NUMBERS			
		TIME	TEMPERATURE	X OR R	Y OR TH
1	0.357000D 05	1	0	0	0

***** SUMMARY OF BOUNDARY DATA *****

----- OF NLPAL -----			-- TEMPERATURE --		----- HEAT TRANSFER COEFFICIENTS -----				
			INFORMATION		RELATED FUNCTION NUMBERS				
NO.	TYPE	FCT FLAG	TEMPERATURE & TIME FCT	ASSOC. FCTS	FORCED CONV.	RADIATION	NATURAL CONV	EXPONENT	FLUX
1	1	1	1.000000D 00		2.910000D 04	0.0	0.0	0.0	0.0
			-3	TIME	-2	0	0	0	0

GRUSS LATTICES AND NUMBERS OF INCREMENTS

R OR X	0.000001	9.810000	13.670000
	5	1	
THETA OR Y	0.0	3.141590	
	10		
Z	0.0	3.900000	19.500000
	1	6	1
			23.420000
			25.340000

Fig. 3.9.2.1. (Contd.)

LISTING OF TABLE FUNCTIONS			
TABLE NUMBER	1	NUMBER OF PAIRS -	3
ARGUMENT	VALUE		
0.0	6.770000000 02		
5.600000000-03	1.200000000 01		
1.000000000 01	1.200000000 01		
TABLE NUMBER	2	NUMBER OF PAIRS -	16
ARGUMENT	VALUE		
0.0	11.84150000 04		
1.000000000-06	0.0		
5.740000000-02	0.0		
5.300000000-02	1.44479000 02		
1.200000000-01	5.12617000 02		
2.455000000-01	5.46744000 02		
5.028000000-01	3.76250000 02		
1.000000000 00	5.87440000 02		
1.500000000 00	5.44624000 02		
2.040000000 00	6.00043000 02		
2.104000000 00	1.07910000 02		
3.030000000 00	6.07024000 02		
3.510000000 00	6.07110000 02		
4.030000000 00	6.17938000 02		
4.150000000 00	6.11450000 02		
1.000000000 02	6.13345000 02		
TABLE NUMBER	3	NUMBER OF PAIRS -	16
ARGUMENT	VALUE		
0.0	7.710000000 02		
8.100000000-02	7.14000000 02		
8.100000000-02	7.14000000 02		
1.032000000-01	4.11000000 02		
2.553000000-01	2.40000000 02		
5.443000000-01	2.62000000 02		
1.040000000 00	2.44000000 02		
1.500000000 00	2.47000000 02		
2.030000000 00	2.35000000 02		
2.430000000 00	2.31000000 02		
2.750000000 00	2.24000000 02		
3.150000000 00	2.24000000 02		
3.550000000 00	2.24000000 02		
4.050000000 00	2.21000000 02		
4.150000000 00	2.20000000 02		
1.000000000 01	2.22000000 02		
TABLE NUMBER	4	NUMBER OF PAIRS -	16
ARGUMENT	VALUE		
0.0	0.0		
3.894000000 00	0.0		
3.403000000 00	5.265000000-01		
6.444000000 00	5.205000000-01		
6.503000000 00	1.144400000 02		
9.044000000 00	1.149400000 02		
9.100000000 00	1.326700000 00		
1.164400000 01	1.326700000 00		

Fig. 3.9.2.1. (Contd.)

1.17000000D 01	1.07870000D 00
1.42990000D 01	1.07870000D 00
1.43000000D 01	9.09600000D-01
1.66990000D 01	9.09600000D-01
1.67000000D 01	7.14500000D-01
1.94990000D 01	7.14500000D-01
1.95000000D 01	0.0

TABLE OF OUTPUT TIMES

OUTPUT NO.	OUTPUT TIME	OUTPUT NO.	OUTPUT TIME	OUTPUT NO.	OUTPUT TIME	OUTPUT NO.	OUTPUT TIME
1	8.330000-02						
2	2.500000-01						
3	5.000000-01						
4	6.000000-01						
5	7.500000-01						
6	1.000000 00						
7	1.500000 00						
8	2.000000 00						
9	2.500000 00						
10	3.000000 00						
11	3.500000 00						
12	4.000000 00						
13	4.500000 00						
14	5.000000 00						
15	5.500000 00						
16	6.000000 00						
17	7.000000 00						
18	8.000000 00						
19	9.000000 00						
20	1.000000 01						

SINCE THIS PROBLEM INVOLVES COOLANT FLOW THE TEMPERATURES WILL BE ADJUSTED TO RANKINE BY ADDING 459.690

Fig. 3.9.2.1. (Contd.)

FINE LATTICE, X OR R, Y OR THETA, AND Z
RADIAL NODE GRID LINES

1	0.000001	2	2.180001	3	4.360001	4	6.540000	5	8.720000
6	13.670000								

RADIAL CELL EDGE GRID LINES

1	0.000001	2	1.090001	3	3.270001	4	5.450000	5	7.630000
6	9.810000	7	13.670000						

1	0.0	2	0.314159	3	0.628318	4	0.942477	5	1.256636
6	1.570795	7	1.884954	8	2.199113	9	2.513272	10	2.827431
11	3.141590								

AXIAL NODE GRID LINES

1	0.0	2	5.200000	3	7.800000	4	10.400000	5	13.000000
6	15.600000	7	18.200000	8	21.400000	9	25.340000		

AXIAL CELL EDGE GRID LINES

1	0.0	2	3.900000	3	6.500000	4	9.100000	5	11.700000
6	14.300000	7	16.900000	8	19.500000	9	23.420000	10	25.340000

LISTING OF ANALYTIC FUNCTIONS

$$F(V) = A(1) + A(2)*V + A(3)*V**2 + A(4)*COS(A(5)*V) + A(6)*EXP(A(7)*V) + A(8)*SIN(A(9)*V) + A(10)*LOG(A(11)*V)$$

NO.	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	A(11)
ANALYTIC FUNCTION NO.	1										
ANALYTIC FUNCTION NO.	2										

Fig. 3.9.2.1. (Contd.)

*****HTGR INPUT DATA SUMMARY*****
 NUMBER OF REFUELING REGIONS 37
 UPPER PLENUM PRESSURE TEMPERATURE COOLANT MGL. WT. MASS FLUX DIAMETERS
 1.00000 1.00000 4.00000 0.96500 0.52097D-010.41670D-01
 TIME FUNCTN NO. -1 -3 0 -2
 ***** SUMMARY OF REFUELING REGION DATA *****

REGION	NUMBER OF CHANNELS	RADIAL POWER FTR	FRACTION OF FULL REGION	ORIFICE AREA	FLOW RATE
SMALL	LARGE				
1	41	664	0.338330	1.000000	1.00000069000.000000
2	41	664	1.127400	1.000000	1.00000059000.000000
3	41	664	1.166000	1.000000	1.00000069000.000000
4	41	664	1.074500	1.000000	1.00000069000.000000
5	41	664	0.796400	1.000000	1.00000069000.000000
6	41	664	0.713100	1.000000	1.00000069000.000000
7	41	664	1.014000	1.000000	1.00000059000.000000
8	41	664	0.995200	1.000000	1.00000059000.000000
9	41	664	0.956700	1.000000	1.00000059000.000000
10	41	664	0.771400	1.000000	1.00000069000.000000
11	41	664	1.483500	1.000000	1.00000059000.000000
12	41	664	1.027500	1.000000	1.00000059000.000000
13	41	664	1.233600	1.000000	1.00000069000.000000
14	41	664	1.237800	1.000000	1.00000069000.000000
15	41	664	1.249200	1.000000	1.00000069000.000000
16	41	664	1.539700	1.000000	1.00000069000.000000
17	41	664	0.890100	1.000000	1.00000069000.000000
18	41	664	1.124300	1.000000	1.00000069000.000000
19	41	664	0.854700	1.000000	1.00000069000.000000
20	29	460	0.449500	0.714286	1.00000069000.000000
21	41	664	0.520500	1.000000	1.00000059000.000000
22	41	664	0.537200	1.000000	1.00000069000.000000
23	29	460	0.494900	0.714286	1.00000069000.000000
24	41	664	0.801600	1.000000	1.00000069000.000000
25	41	664	1.526100	1.000000	1.00000069000.000000
26	29	460	1.114000	0.714286	1.00000069000.000000
27	41	664	1.233800	1.000000	1.00000059000.000000
28	41	664	0.556900	1.000000	1.00000069000.000000
29	29	460	0.865600	0.714286	1.00000069000.000000
30	41	664	1.784300	1.000000	1.00000069000.000000
31	41	664	1.112900	1.000000	1.00000059000.000000
32	29	460	0.853500	0.714286	1.00000069000.000000
33	41	664	1.354400	1.000000	1.00000059000.000000
34	41	664	0.780800	1.000000	1.00000069000.000000
35	29	460	0.427200	0.714286	1.00000069000.000000
36	41	664	1.062900	1.000000	1.00000069000.000000
37	41	664	1.404300	1.000000	1.00000069000.000000

PLEASE NOTE THAT COOLANT FLOW IS ASSUMED TO BE IN THE POSITIVE Z DIRECTION.
 THEREFOR THE +Z AXIS FOR HTGR REACTORS MUST BE FROM THE UPPER PLENUM TO LOWER PLENUM.

LOOSE SUPPORT BLOCK DATA FOR HTGR REACTORS
 DIA OF COLLECTOR(FT)= 0.62917 DIA OF LOWER PLENUM(FT)= 1.75000
 DISTANCE FROM TOP OF MODEL TO FIRST EXPANSION= 22.92000
 DISTANCE FROM TOP OF MODEL TO SECOND EXPANSION= 24.17000

ORIFICE AND CHANNEL DATA FOR THE HTGR REACTORS
 ORIFICE MAX AREA MIN AREA LENGTH DIA.
 1.14583 0.750000D-010.968750 0.623295

UPPER REFLECTOR ELEMENT PLENUM AREA 8.07640

Fig. 3.9.2.1. (Contd.)

LOWER EXPANSION AREAS FIRST SECOND
 1.84030 2.40300

THIS PROBLEM CONTAINS 765 NODES.

K=	JZNRM=	DZCUL=	DZEPLM=	NUPZEK=
1	3.90000	0.0	0.0	2
2	2.60000	0.0	0.0	3
3	2.60000	0.0	0.0	4
4	2.60000	0.0	0.0	5
5	2.60000	0.0	0.0	6
6	2.60000	0.0	0.0	7
7	2.60000	0.0	0.0	8
8	3.42000	0.50000	0.0	9
9	0.0	0.75000	1.17000	10

THE SPECIFIC HEAT FOR THE MATERIALS DENOTED BELOW WILL BE EVALUATED ACCORDING TO THE FOLLOWING ALGORITHMS.

NUMBER	ALGORITHM
1	$(\text{RHO} \cdot \text{CP})_{\text{FUEL}} = 3.38 + 2.74\text{E-}2 \cdot T - 4.88\text{E-}6 \cdot T^2 - 0.146\text{E+}6 / T^2$ $(\text{CP})_{\text{MODERATOR}} = -0.103 + 6.472\text{E-}4 \cdot T - 2.667\text{E-}7 \cdot T^2 + 3.97\text{E-}11 \cdot T^3$ MODERATOR DENSITY= 107.0000 THE SPECIFIC HEAT OF THE NODE IS A FUNCTION OF THE PCACFE OF THE REGION SUCH THAT $\text{CP} \cdot \text{RHG} = (\text{RHO} \cdot \text{CP} \cdot \text{VOLF}(\text{FUEL}) + \text{RHO} \cdot \text{CP} \cdot \text{VOLF}(\text{MODERATOR})) \cdot \text{PCACFE} + (1 - \text{PCACFE}) \cdot \text{RHO} \cdot \text{CP} \cdot \text{VOLF}(\text{MODERATOR})$ WHERE VOLF = VOLUME FRACTION OF THAT MATERIAL
2	$\text{CP} = -0.103 + 6.472\text{E-}4 \cdot T - 2.667\text{E-}7 \cdot T^2 + 3.97\text{E-}11 \cdot T^3$

THE INITIAL TIME INCREMENT = 1.000000D-01
 AFTER EACH TIME STEP THE TIME INCREMENT WILL BE MULTIPLIED BY 1.200000D 00
 THIS TIME STEP FUNCTION WILL BE APPLIED UNTIL TIME = 1.000000D 01

Fig. 3.9.2.1. (Contd.)

STABILITY CRITERION FOR EACH NODE

1	7.51740-03	2	7.51740-03	3	7.51740-03	4	7.51740-03	5	7.51740-03	6	7.51740-03
7	7.51740-03	8	7.51740-03	9	7.51740-03	10	7.51740-03	11	7.51740-03	12	7.51740-03
13	7.51740-03	14	7.51740-03	15	7.51740-03	16	7.51740-03	17	7.51740-03	18	7.51740-03
19	7.51740-03	20	1.15230-02	21	7.51740-03	22	7.51740-03	23	1.15230-02	24	7.51740-03
25	7.51740-03	26	1.15230-02	27	7.51740-03	28	7.51740-03	29	1.15230-02	30	7.51740-03
31	7.51740-03	32	1.15230-02	33	7.51740-03	34	7.51740-03	35	1.15230-02	36	7.51740-03
37	7.51740-03	38	2.20970 00	39	3.33480 00	40	3.39600 00	41	3.33480 00	42	2.20970 00
43	3.33480 00	44	3.39600 00	45	3.33480 00	46	2.20970 00	47	3.33480 00	48	3.39600 00
49	3.33480 00	50	2.20970 00	51	3.33480 00	52	3.39600 00	53	3.33480 00	54	2.20970 00
55	3.33480 00	56	3.39600 00	57	3.33480 00	58	2.20970 00	59	3.33480 00	60	3.39600 00
61	3.33480 00	62	6.07550-01	63	9.38080-01	64	1.25290 00	65	9.38080-01	66	6.07550-01
67	9.38080-01	68	1.25290 00	69	9.38080-01	70	6.07550-01	71	9.38080-01	72	1.25290 00
73	9.38080-01	74	6.07550-01	75	9.38080-01	76	1.25290 00	77	9.38080-01	78	6.07550-01
79	9.38080-01	80	1.25290 00	81	9.38080-01	82	6.07550-01	83	9.38080-01	84	1.25290 00
85	9.38080-01	86	7.08720-03	87	7.08720-03	88	7.08720-03	89	7.08720-03	90	7.08720-03
91	7.08720-03	92	7.08720-03	93	7.08720-03	94	7.08720-03	95	7.08720-03	96	7.08720-03
97	7.08720-03	98	7.08720-03	99	7.08720-03	100	7.08720-03	101	7.08720-03	102	7.08720-03
103	7.08720-03	104	7.08720-03	105	1.10590-02	106	7.08720-03	107	7.08720-03	108	1.10590-02
109	7.08720-03	110	7.08720-03	111	1.10590-02	112	7.08720-03	113	7.08720-03	114	1.10590-02
115	7.08720-03	116	7.08720-03	117	1.10590-02	118	7.08720-03	119	7.08720-03	120	1.10590-02
121	7.08720-03	122	7.08720-03	123	2.84040 00	124	5.01560 00	125	5.15550 00	126	5.01560 00
127	2.84040 00	128	5.01560 00	129	5.15550 00	130	5.01560 00	131	2.84040 00	132	5.01560 00
133	5.15550 00	134	5.01560 00	135	2.84040 00	136	5.01560 00	137	5.15550 00	138	5.01560 00
139	2.84040 00	140	5.01560 00	141	5.15550 00	142	5.01560 00	143	2.84040 00	144	5.01560 00
145	5.15550 00	146	5.01560 00	147	6.47060-01	148	1.03570 00	149	1.43330 00	150	1.03570 00
151	6.47060-01	152	1.03570 00	153	1.43330 00	154	1.03570 00	155	6.47060-01	156	1.03570 00
157	1.43330 00	158	1.03570 00	159	6.47060-01	160	1.03570 00	161	1.43330 00	162	1.03570 00
163	6.47060-01	164	1.03570 00	165	1.43330 00	166	1.03570 00	167	6.47060-01	168	1.03570 00
169	1.43330 00	170	1.03570 00	171	7.08580-03	172	7.08580-03	173	7.08580-03	174	7.08580-03
175	7.08580-03	176	7.08580-03	177	7.08580-03	178	7.08580-03	179	7.08580-03	180	7.08580-03
181	7.08580-03	182	7.08580-03	183	7.08580-03	184	7.08580-03	185	7.08580-03	186	7.08580-03
187	7.08580-03	188	7.08580-03	189	7.08580-03	190	1.10560-02	191	7.08580-03	192	7.08580-03
193	1.10560-02	194	7.08580-03	195	7.08580-03	196	1.10560-02	197	7.08580-03	198	7.08580-03
199	1.10560-02	200	7.08580-03	201	7.08580-03	202	1.10560-02	203	7.08580-03	204	7.08580-03
205	1.10560-02	206	7.08580-03	207	7.08580-03	208	2.67480 00	209	4.52140 00	210	4.63480 00
211	4.52140 00	212	2.67480 00	213	4.52140 00	214	4.63480 00	215	4.52140 00	216	2.67480 00
217	4.52140 00	218	4.63480 00	219	4.52140 00	220	2.67480 00	221	4.52140 00	222	4.63480 00
223	4.52140 00	224	2.67480 00	225	4.52140 00	226	4.63480 00	227	4.52140 00	228	2.67480 00
229	4.52140 00	230	4.63480 00	231	4.52140 00	232	6.38060-01	233	1.01290 00	234	1.38990 00
235	1.01290 00	236	6.38060-01	237	1.01290 00	238	1.38990 00	239	1.01290 00	240	6.38060-01
241	1.01290 00	242	1.38990 00	243	1.01290 00	244	6.38060-01	245	1.01290 00	246	1.38990 00
247	1.01290 00	248	6.38060-01	249	1.01290 00	250	1.38990 00	251	1.01290 00	252	6.38060-01
253	1.01290 00	254	1.38990 00	255	1.01290 00	256	7.08580-03	257	7.08580-03	258	7.08580-03
259	7.08580-03	260	7.08580-03	261	7.08580-03	262	7.08580-03	263	7.08580-03	264	7.08580-03
265	7.08580-03	266	7.08580-03	267	7.08580-03	268	7.08580-03	269	7.08580-03	270	7.08580-03
271	7.08580-03	272	7.08580-03	273	7.08580-03	274	7.08580-03	275	1.10560-02	276	7.08580-03
277	7.08580-03	278	1.10560-02	279	7.08580-03	280	7.08580-03	281	1.10560-02	282	7.08580-03
283	7.08580-03	284	1.10560-02	285	7.08580-03	286	7.08580-03	287	1.10560-02	288	7.08580-03
289	7.08580-03	290	1.10560-02	291	7.08580-03	292	7.08580-03	293	2.67480 00	294	4.52140 00
295	4.63480 00	296	4.52140 00	297	2.67480 00	298	4.52140 00	299	4.63480 00	300	4.52140 00
301	2.67480 00	302	4.52140 00	303	4.63480 00	304	4.52140 00	305	2.67480 00	306	4.52140 00
307	4.63480 00	308	4.52140 00	309	2.67480 00	310	4.52140 00	311	4.63480 00	312	4.52140 00
313	2.67480 00	314	4.52140 00	315	4.63480 00	316	4.52140 00	317	6.38060-01	318	1.01290 00
319	1.38990 00	320	1.01290 00	321	6.38060-01	322	1.01290 00	323	1.38990 00	324	1.01290 00
325	6.38060-01	326	1.01290 00	327	1.38990 00	328	1.01290 00	329	6.38060-01	330	1.01290 00
331	1.38990 00	332	1.01290 00	333	6.38060-01	334	1.01290 00	335	1.38990 00	336	1.01290 00
337	6.38060-01	338	1.01290 00	339	1.38990 00	340	1.01290 00	341	7.08580-03	342	7.08580-03
343	7.08580-03	344	7.08580-03	345	7.08580-03	346	7.08580-03	347	7.08580-03	348	7.08580-03

Fig. 3.9.2.1. (Contd.)

709	1.4524D-01	710	1.0922D-01	711	1.0922D-01	712	1.4524D-01	713	1.0922D-01	714	1.0922D-01
715	1.4524D-01	716	1.0922D-01	717	1.0922D-01	718	3.0652D 00	719	5.7620D 00	720	5.9474D 00
721	5.7620D 00	722	3.0652D 00	723	5.7620D 00	724	5.9474D 00	725	5.7620D 00	726	3.0652D 00
727	5.7620D 00	728	5.9474D 00	729	5.7620D 00	730	3.0652D 00	731	5.7620D 00	732	5.9474D 00
733	5.7620D 00	734	3.0652D 00	735	5.7620D 00	736	5.9474D 00	737	5.7620D 00	738	3.0652D 00
739	5.7620D 00	740	5.9474D 00	741	5.7620D 00	742	6.5806D-01	743	1.0642D 00	744	1.4884D 00
745	1.0642D 00	746	6.5806D-01	747	1.0642D 00	748	1.4884D 00	749	1.0642D 00	750	6.5806D-01
751	1.0642D 00	752	1.4884D 00	753	1.0642D 00	754	6.5806D-01	755	1.0642D 00	756	1.4884D 00
757	1.0642D 00	758	6.5806D-01	759	1.0642D 00	760	1.4884D 00	761	1.0642D 00	762	6.5806D-01
763	1.0642D 00	764	1.4884D 00	765	1.0642D 00						

THE STABILITY CRITERION IS 7.0859111D-03 FOR POINT 462

Fig. 3.9.2.1. (Contd.)

3.9.3. *Temperature Map*

HEXEREI will print out the solid and coolant temperatures for all nodes in the model if requested by the input data (see section 3.6.3(e)). The temperatures are printed in a series of two-dimensional arrays, each presenting one axial plane. The maximum temperature occurring within and the axial coordinate for each axial plane are printed next to the two-dimensional array representing the nodal temperatures within that plane. In addition, the location of the maximum temperature is notated by asterisks. If any of the temperatures in that plane exceed the maximum of 9999.9 or are exactly equal to the maximum, then more than one field will be filled with asterisks. The maximum temperature for all of the axial planes and the plane in which it occurs are printed out at the start of the series of arrays and repeated at the top of each new page. The above information is first presented for the solid temperatures if requested and then this same format is used to present the coolant temperatures if requested. The coolant exit temperatures are also printed in the same format, but only one plane is needed.

3.9.4. *Summary Information*

To aid in interpreting the mass of numbers produced by the code, a summary output was developed (Fig. 3.9.4.1). The basic two-dimensional array format is used to present the summary information that is reported for each refueling region. This information includes the coolant flow rate, direction, solid maximum temperature and location, coolant exit temperature, volume averaged solid temperature, heat retained, heat lost to coolant, and heat generated. If the problem does not utilize the coolant flow option, then some of the above information is omitted and output in single two-dimensional array. If not, then two arrays will be required.

In addition, some overall and specialized information is also presented. Most of the information has self-explanatory labels. The four that perhaps require additional explanation are: 1) Coolant mean temperature represents the mass flow rate weighted exit temperature; 2) Total heat transferred to the coolant represents the sum of all heat transferred from the solid to the coolant; 3) Number labeled "net" immediately following refers to the change of coolant heat retained from the top to bottom plenums; 4) Active core volume average temperature refers to the active core of HTGRs.

3.9.5. *Steady-State Temperature Distribution*

For steady-state calculations, the program writes "STEADY-STATE TEMPERATURE DISTRIBUTION" and indicates the number of steady-state iterations completed and the time at which the steady-state calculations were performed. The temperature distribution is then presented in map form as discussed above. The boundary temperatures are then listed in a table along with a message indicating the status of the calculation. The elapsed computer time, measured in seconds from the start of the job, is also printed out.

3.9.6. *Transient Temperature Distribution*

For the transient calculation, the program writes "TRANSIENT TEMPERATURE DISTRIBUTION" and indicates the number of time steps completed for each specified time and the time at which the temperature distribution is written. The temperature distribution is listed in the same manner as for steady-state problems.

3.9.7. *Special Monitoring of Temperatures*

One may wish to follow the temperatures of a few of the nodes during the calculations. As optional output, one may tabulate temperatures of specified nodes as a function of the number of iterations for steady-state problems or the number of time steps for transient problems. For more details on this option, see section 3.6.4(d).

SOLID NODE TEMPERATURES

DATE = 10-07-76, JOB NO./NAME = 0/GEG012

*****FT.ST.VRIAN DBDA2 IMPLICIT 9-3-76*****

MAX. TEMP. = 1714. AT LEVEL = 7

LEVEL = 4

MAX. TEMP. = 1583., Z = 10.40

```
      823.      818.      815.      818.      823.
      912.      1014.      1044.      1017.      912.
      818.1012.      1309.      1445.      1511.      1312.      1008.      817.
      815.1042.      1391.      1460.      1409.      1435.      1339.      1030.      813.
      819.1021.      1502.      1416.      1439.      1461.      1428.      1342.      1008.      817.      823.
      918.      1376.      1540.      1381.      1307.      1469.      1392.      1319.      912.
      825.      819.1021.      1456.      1486.      1397.      1452.      1530.      1395.      1013.      818.
      816.1049.      *****      1483.      1482.      1443.      1535.      1044.      815.
      820.1024.      1376.      1348.      1479.      1413.      1024.      820.
      917.      1015.      1040.      1022.      920.
      825.      818.      814.      820.      825.
```

LEVEL = 5

MAX. TEMP. = 1653., Z = 13.00

```
      833.      828.      823.      828.      833.
      942.      1064.      1097.      1066.      942.
      827.1062.      1409.      1533.      1590.      1412.      1058.      827.
      823.1096.      1485.      1547.      1502.      1525.      1439.      1085.      822.
      829.1070.      1582.      1508.      1528.      1548.      1518.      1442.      1059.      827.      833.
      947.      1467.      1617.      1477.      1410.      1554.      1487.      1418.      942.
      835.      829.1070.      1543.      1569.      1491.      1540.      1608.      1489.      1063.      828.
      824.1102.      *****      1567.      1566.      1532.      1611.      1097.      823.
      829.1073.      1467.      1446.      1563.      1499.      1073.      829.
      946.      1064.      1094.      1071.      948.
      835.      828.      823.      829.      836.
```

LEVEL = 6

MAX. TEMP. = 1698., Z = 15.60

```
      841.      835.      830.      835.      841.
      964.      1102.      1137.      1103.      964.
      834.1100.      1492.      1597.      1645.      1494.      1097.      834.
      829.1136.      1556.      1609.      1572.      1591.      1517.      1127.      828.
      835.1106.      1638.      1577.      1594.      1610.      1586.      1520.      1097.      834.      841.
      968.      1540.      1668.      1550.      1494.      1616.      1559.      1499.      964.
      843.      836.1106.      1605.      1628.      1563.      1603.      1661.      1560.      1101.      835.
      830.1141.      *****      1626.      1625.      1597.      1662.      1137.      830.
      836.1109.      1541.      1524.      1622.      1567.      1109.      836.
      968.      1102.      1134.      1107.      970.
      843.      835.      829.      836.      843.
```

Fig. 3.9.4.1. Summary Output from HEXERE12

SOLID NODE TEMPERATURES

DATE = 10-07-76, JOB NO./NAME = 0/GFG012

*****FT.ST.VRIAN DBDA2 IMPLICIT 9-3-76*****

MAX. TEMP. = 1714. AT LEVEL = 7

LEVEL = 7

MAX. TEMP. = 1714., Z = 18.20

```

      846.      838.      833.      838.      846.
      977.      1122.      1158.      1123.      977.
      838.1121.      1549.      1634.      1673.      1551.      1118.      838.
      833.1157.      1602.      1645.      1615.      1631.      1571.      1151.      832.
      839.1125.      1667.      1619.      1633.      1646.      1626.      1573.      1119.      838.      846.
      980.      1588.      1692.      1598.      1553.      1650.      1605.      1555.      977.
      847.      839.1125.      1641.      1660.      1608.      1640.      1686.      1604.      1121.      838.
      833.1160.      *****      1659.      1658.      1635.      1686.      1158.      833.
      839.1127.      1588.      1576.      1654.      1609.      1127.      839.
      980.      1122.      1156.      1126.      981.
      847.      838.      832.      839.      847.

```

LEVEL = 8

MAX. TEMP. = 1504., Z = 21.46

```

      843.      836.      830.      836.      843.
      968.      1105.      1137.      1105.      967.
      836.1105.      1471.      1492.      1498.      1472.      1105.      836.
      830.1137.      1486.      1497.      1491.      1494.      1480.      1136.      830.
      836.1106.      1497.      1492.      1495.      1497.      1493.      1480.      1105.      836.      843.
      968.      1477.      1504.      1488.      1479.      1498.      1489.      1472.      967.
      843.      836.1106.      1493.      1499.      1490.      1496.      1503.      1487.      1105.      836.
      830.1138.      *****      1499.      1499.      1495.      1500.      1137.      830.
      836.1106.      1477.      1481.      1495.      1480.      1106.      836.
      968.      1105.      1137.      1106.      968.
      843.      836.      830.      836.      843.

```

LEVEL = 9

MAX. TEMP. = 1470., Z = 25.34

```

      842.      835.      829.      835.      842.
      965.      1101.      1133.      1101.      965.
      835.1101.      1455.      1465.      1465.      1455.      1101.      835.
      829.1133.      1464.      1470.      1470.      1470.      1463.      1132.      829.
      835.1102.      1465.      1470.      1470.      1470.      1470.      1463.      1101.      835.      842.
      966.      1458.      1470.      1470.      1470.      1470.      1470.      1456.      965.
      842.      835.1102.      1465.      1470.      1470.      1470.      *****      1464.      1101.      835.
      829.1134.      1466.      1470.      1470.      1470.      1465.      1133.      829.
      835.1102.      1458.      1463.      1465.      1459.      1102.      835.
      965.      1101.      1133.      1102.      966.
      842.      835.      829.      835.      842.

```

Fig. 3.9.4.1. (Contd.)

```
COOLANT NODE TEMPERATURES
```

MAX. TEMP. = 2169. AT LEVEL = 7

	*****	*****	*****	*****	*****	*****	*****
	*****	*****	*****	*****	*****	*****	*****
	*****	1107.	1047.	1043.	1105.	*****	*****
	*****	1057.	1045.	1053.	1048.	1077.	*****
	*****	1043.	1051.	1048.	1045.	1049.	1075.
	*****	1076.	1042.	1060.	1103.	1045.	1057.
	*****	1046.	1044.	1055.	1046.	1042.	1056.
	*****	1043.	1044.	1044.	1047.	1043.	*****
	*****	1076.	1073.	1044.	1069.	*****	*****
	*****	*****	*****	*****	*****	*****	*****
	*****	*****	*****	*****	*****	*****	*****

[illegible]

	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.
	1460.1460.	1550.	1630.	1679.	1551.
	1460.1460.	1593.	1640.	1605.	1623.
	1460.1460.	1672.	1609.	1625.	1641.
	1460.	1588.	1702.	1587.	1550.
1460.1460.1460.	1638.	1659.	1597.	1634.	1694.
	1460.1460.	****	1657.	1657.	1628.
	1460.1460.	1588.	1568.	1655.	1614.
	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.

COOLANT NODE TEMPERATURES

DATE = 10-07-76, JOB NO./NAME = 0/GEG012

*****FT,ST,VRIAN DBDA2 IMPLICIT 9-3-76*****

MAX. TEMP. = 2169. AT LEVEL = 7

LEVEL = 4

MAX. TEMP. = 1980., Z = 10.40

	1460.	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.	1460.
	1460.1460.	1728.	1847.	1909.	1731.	1460.1460.
	1460.1460.	1797.	1861.	1814.	1838.	1754.
	1460.1460.	1901.	1820.	1841.	1862.	1831.
	1460.	1782.	1938.	1789.	1730.	1869.
	1460.1460.1460.	1857.	1885.	1803.	1853.	1928.
	1460.1460.	****	1882.	1881.	1845.	1933.
	1460.1460.	1783.	1761.	1879.	1816.	1460.1460.
	1460.	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.	1460.

LEVEL = 5

MAX. TEMP. = 2092., Z = 13.00

	1460.	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.	1460.
	1460.1460.	1846.	1967.	2026.	1848.	1460.1460.
	1460.1460.	1919.	1981.	1936.	1959.	1874.
	1460.1460.	2018.	1942.	1962.	1982.	1952.
	1460.	1900.	2054.	1911.	1849.	1989.
	1460.1460.1460.	1977.	2004.	1925.	1974.	2044.
	1460.1460.	****	2002.	2001.	1966.	2048.
	1460.1460.	1901.	1881.	1998.	1933.	1460.1460.
	1460.	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.	1460.

LEVEL = 6

MAX. TEMP. = 2145., Z = 15.60

	1460.	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.	1460.
	1460.1460.	1933.	2039.	2089.	1935.	1460.1460.
	1460.1460.	1997.	2052.	2013.	2033.	1959.
	1460.1460.	2082.	2018.	2035.	2052.	2027.
	1460.	1980.	2113.	1991.	1937.	2058.
	1460.1460.1460.	2047.	2071.	2003.	2045.	2105.
	1460.1460.	****	2069.	2068.	2039.	2108.
	1460.1460.	1980.	1965.	2065.	2007.	1460.1460.
	1460.	1460.	1460.	1460.	1460.	1460.
	1460.	1460.	1460.	1460.	1460.	1460.

Fig. 3.9.4.1. (Contd.)

COOLANT NODE TEMPERATURES

DATE = 10-07-76, JOB NO./NAME = 0/GEG012

*****FT.ST.VRIAN DBDA2 IMPLICIT 9-3-76*****

MAX. TEMP. = 2169. AT LEVEL = 7

LEVEL = 7

MAX. TEMP. = 2169., Z = 18.20

1460.	1460.	1460.	1460.	1460.	1460.	1460.
1460.	1460.	1460.	1460.	1460.	1460.	1460.
1460.1460.	1996.	2084.	2124.	1998.	1460.1460.	
1460.1460.	2050.	2095.	2063.	2080.	2018.	1460.1460.
1460.1460.	2119.	2067.	2082.	2096.	2075.	2020.
1460.	2034.	2145.	2046.	2001.	2100.	2052.
1460.1460.1460.	2090.	2111.	2056.	2090.	2138.	2052.
1460.1460.	*****	2109.	2109.	2084.	2139.	1460.1460.
1460.1460.	2035.	2023.	2105.	2056.	1460.1460.	
1460.	1460.	1460.	1460.	1460.	1460.	
1460.	1460.	1460.	1460.	1460.	1460.	

LEVEL = 8

MAX. TEMP. = 2008., Z = 21.46

1460.	1460.	1460.	1460.	1460.	1460.	1460.
1460.	1460.	1460.	1460.	1460.	1460.	1460.
1460.1460.	1944.	1981.	1995.	1945.	1460.1460.	
1460.1460.	1969.	1987.	1976.	1982.	1956.	1460.1460.
1460.1460.	1993.	1977.	1983.	1988.	1980.	1956.
1460.	1960.	2003.	1969.	1949.	1989.	1972.
1460.1460.1460.	1984.	1993.	1973.	1986.	2001.	1970.
1460.1460.	*****	1992.	1992.	1984.	1999.	1460.1460.
1460.1460.	1960.	1958.	1989.	1967.	1460.1460.	
1460.	1460.	1460.	1460.	1460.	1460.	
1460.	1460.	1460.	1460.	1460.	1460.	

LEVEL = 9

MAX. TEMP. = 1965., Z = 25.34

1460.	1460.	1460.	1460.	1460.	1460.	1460.
1460.	1460.	1460.	1460.	1460.	1460.	1460.
1460.1460.	1930.	1952.	1959.	1931.	1460.1460.	
1460.1460.	1946.	1957.	1951.	1954.	1939.	1460.1460.
1460.1460.	1958.	1952.	1955.	1957.	1953.	1940.
1460.	1937.	1964.	1948.	1938.	1958.	1949.
1460.1460.1460.	1953.	1959.	1950.	1956.	1963.	1946.
1460.1460.	*****	1959.	1959.	1955.	1961.	1460.1460.
1460.1460.	1937.	1940.	1956.	1940.	1460.1460.	
1460.	1460.	1460.	1460.	1460.	1460.	
1460.	1460.	1460.	1460.	1460.	1460.	

Fig. 3.9.4.1. (Contd.)

RESULTS FROM TRANSIENT CALCULATIONS
 TIME = 2.50000D-01 TIME STEPS COMPLETED = 0 ELAPSED CPU SECONDS = 36.78
 COOLANT EXIT TEMPERATURES DATE = 10-07-76, JOB NO./NAME = 0/GFG012

*****FT.ST.VRIAN DBDA2 IMPLICIT 9-3-76*****

MAX. TEMP. = 1504. AT LEVEL = 1

LEVEL = 1

MAX. TEMP. = 1504., Z = 0.0

1000.	1000.	1000.	1000.	1000.	1000.
1000.	1000.	1000.	1000.	1000.	1000.
1000.1000.	1470.	1492.	1498.	1470.	1000.1000.
1000.1000.	1485.	1496.	1491.	1494.	1479.
1000.1000.	1497.	1492.	1494.	1496.	1493.
1000.	1477.	1504.	1488.	1478.	1497.
1000.1000.1000.	1493.	1499.	1490.	1496.	1503.
1000.1000.	1477.	1498.	1498.	1495.	1500.
1000.1000.	1477.	1480.	1495.	1480.	1000.1000.
1000.	1477.	1480.	1495.	1480.	1000.1000.
1000.	1477.	1480.	1495.	1480.	1000.1000.
1000.	1477.	1480.	1495.	1480.	1000.1000.

Fig. 3.9.4.1. (Contd.)

RESULTS FROM TRANSIENT CALCULATIONS
 TIME = 2.500000-01 TIME STEPS COMPLETED = 0 ELAPSED CPU SECONDS= 35.01
 SUMMARY OUTPUT DATE = 10-05-76, JOB NO./NAME = 07GEG012

*****FT.ST.VRIAN D4DA2 IMPLICIT 9-3-76*****

TIME= 0.25000 HOURS, TOTAL FLOW= 27275.6 LB/HQ PRESSURE DROP(PSI)= 0.1312
 EXIT MEAN TEMPERATURE 1491.4269 UPPER PLEN. TEMP= 320.4201
 MAX SOLID TEMP= 1714.2577, MAX EXIT TEMP= 1503.6067

REGION 82	REGION 83	REGION 84	REGION 85	REGION 62
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 846.	SOLD= 838.	SOLD= 833.	SOLD= 838.	SOLD= 846.
HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 0.	FLO= 0.	FLO= 0.

REGION 58	REGION 59	REGION 60	REGION 61	REGION 38
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 977.	SOLD= 1122.	SOLD= 1158.	SOLD= 1123.	SOLD= 977.
HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 0.	FLO= 0.	FLO= 0.

REGION 81	REGION 57	REGION 35	REGION 36	REGION 37	REGION 20	REGION 39	REGION 63
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 838.	SOLD= 1121.	SOLD= 1549.	SOLD= 1634.	SOLD= 1673.	SOLD= 1551.	SOLD= 1118.	SOLD= 838.
HEOT= 1000.	HEOT= 1000.	HEOT= 1470.	HEOT= 1492.	HEOT= 1498.	HEOT= 1470.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 398.	FLO= 821.	FLO= 846.	FLO= 410.	FLO= 0.	FLO= 0.

REGION 80	REGION 56	REGION 34	REGION 18	REGION 19	REGION 8	REGION 21	REGION 40	REGION 64
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 833.	SOLD= 1157.	SOLD= 1602.	SOLD= 1645.	SOLD= 1615.	SOLD= 1631.	SOLD= 1571.	SOLD= 1151.	SOLD= 832.
HEOT= 1000.	HEOT= 1000.	HEOT= 1465.	HEOT= 1496.	HEOT= 1491.	HEOT= 1494.	HEOT= 1479.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 756.	FLO= 827.	FLO= 777.	FLO= 809.	FLO= 632.	FLO= 0.	FLO= 0.

REGION 79	REGION 55	REGION 33	REGION 17	REGION 7	REGION 2	REGION 9	REGION 22	REGION 41	REGION 65	REGION 66
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 839.	SOLD= 1125.	SOLD= 1667.	SOLD= 1619.	SOLD= 1633.	SOLD= 1646.	SOLD= 1625.	SOLD= 1573.	SOLD= 1119.	SOLD= 838.	SOLD= 846.
HEOT= 1000.	HEOT= 1000.	HEOT= 1497.	HEOT= 1492.	HEOT= 1494.	HEOT= 1496.	HEOT= 1493.	HEOT= 1479.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 844.	FLO= 786.	FLO= 812.	FLO= 827.	FLO= 801.	FLO= 643.	FLO= 0.	FLO= 0.	FLO= 0.

REGION 54	REGION 32	REGION 16	REGION 6	REGION 1	REGION 3	REGION 10	REGION 23	REGION 42
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 980.	SOLD= 1588.	SOLD= 1692.	SOLD= 1598.	SOLD= 1553.	SOLD= 1650.	SOLD= 1605.	SOLD= 1555.	SOLD= 977.
HEOT= 1000.	HEOT= 1477.	HEOT= 1504.	HEOT= 1488.	HEOT= 1478.	HEOT= 1497.	HEOT= 1489.	HEOT= 1471.	HEOT= 1000.
FLO= 0.	FLO= 551.	FLO= 845.	FLO= 731.	FLO= 484.	FLO= 831.	FLO= 752.	FLO= 434.	FLO= 0.

REGION 78	REGION 77	REGION 53	REGION 31	REGION 15	REGION 5	REGION 4	REGION 11	REGION 24	REGION 43	REGION 67
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 847.	SOLD= 839.	SOLD= 1125.	SOLD= 1641.	SOLD= 1560.	SOLD= 1608.	SOLD= 1640.	SOLD= 1686.	SOLD= 1604.	SOLD= 1121.	SOLD= 838.
HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1493.	HEOT= 1499.	HEOT= 1490.	HEOT= 1496.	HEOT= 1503.	HEOT= 1486.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 0.	FLO= 827.	FLO= 837.	FLO= 760.	FLO= 821.	FLO= 844.	FLO= 763.	FLO= 0.	FLO= 0.

REGION 76	REGION 52	REGION 30	REGION 14	REGION 13	REGION 12	REGION 25	REGION 44	REGION 68
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 833.	SOLD= 1160.	SOLD=*****	SOLD= 1659.	SOLD= 1658.	SOLD= 1635.	SOLD= 1686.	SOLD= 1158.	SOLD= 833.
HEOT= 1000.	HEOT= 1000.	HEOT=*****	HEOT= 1498.	HEOT= 1498.	HEOT= 1495.	HEOT= 1500.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 842.	FLO= 836.	FLO= 836.	FLO= 813.	FLO= 847.	FLO= 0.	FLO= 0.

Fig. 3.9.4.1. (Contd.)

SUMMARY OUTPUT

DATE = 10-05-76, JOB NO./NAME = 0/GEGJ12

*****FT.ST.VRIAN DBDA2 IMPLICIT 9-3-76*****

TIME= 0.25000 HOURS, TOTAL FLOW= 27275.6 LB/HR PRESSURE DROP(PSI)= 0.1312
 EXIT MEAN TEMPERATURE 1491.4269 UPPER PLEN. TEMP= 320.4201
 MAX SOLID TEMP= 1714.2577, MAX EXIT TEMP= 1503.6067

REGION 75	REGION 51	REGION 29	REGION 28	REGION 27	REGION 26	REGION 45	REGION 69
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 839.	SOLD= 1127.	SOLD= 1588.	SOLD= 1576.	SOLD= 1654.	SOLD= 1609.	SOLD= 1127.	SOLD= 839.
HEOT= 1000.	HEOT= 1000.	HEOT= 1477.	HEOT= 1480.	HEOT= 1495.	HEOT= 1480.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 553.	FLO= 655.	FLO= 838.	FLO= 587.	FLO= 0.	FLO= 0.

REGION 50	REGION 49	REGION 48	REGION 47	REGION 46
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 980.	SOLD= 1122.	SOLD= 1156.	SOLD= 1126.	SOLD= 981.
HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 0.	FLO= 0.	FLO= 0.

REGION 74	REGION 73	REGION 72	REGION 71	REGION 70
MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7	MXLV= 7
SOLD= 847.	SOLD= 838.	SOLD= 832.	SOLD= 839.	SOLD= 847.
HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.	HEOT= 1000.
FLO= 0.	FLO= 0.	FLO= 0.	FLO= 0.	FLO= 0.

Fig. 3.9.4.1. (Contd.)

SUMMARY OUTPUT

DATE = 10-05-76, JOB NO./NAME = 07GEG012

*****FT.ST.VRIAN DBDA2 IMPLICIT 9-3-76*****

TIME= 0.25000 HOURS, TOTAL FLOW= 27275.6 LB/HR PRESSURE DROP(PSI)= 0.1312
 EXIT MEAN TEMPERATURE 1491.4269 UPPER PLEN. TEMP= 320.4201
 MAX SOLID TEMP= 1714.2577, MAX EXIT TEMP= 1503.6067

REGION 82	REGION 83	REGION 84	REGION 85	REGION 62
QC=.0	QC=.0	QC=.0	QC=.0	QC=.0
QS=.173D 07	QS=.195D 07	QS=.278D 07	QS=.195D 07	QS=.173D 07
QG=.0	QG=.0	QG=.0	QG=.0	QG=.0
TVOL= 821.	TVOL= 816.	TVOL= 813.	TVOL= 816.	TVOL= 821.

REGION 58	REGION 59	REGION 60	REGION 61	REGION 38
QC=.0	QC=.0	QC=.0	QC=.0	QC=.0
QS=.417D 07	QS=.737D 07	QS=.770D 07	QS=.739D 07	QS=.417D 07
QG=.0	QG=.0	QG=.0	QG=.0	QG=.0
TVOL= 906.	TVOL= 1002.	TVOL= 1027.	TVOL= 1003.	TVOL= 906.

REGION 81	REGION 57	REGION 35	REGION 36	REGION 37	REGION 20	REGION 39	REGION 63
QC=.0	QC=.0	QC=.568D 06	QC=.119D 07	QC=.124D 07	QC=.586D 06	QC=.0	QC=.0
QS=.195D 07	QS=.736D 07	QS=.902D 07	QS=.906D 07	QS=.938D 07	QS=.904D 07	QS=.733D 07	QS=.195D 07
QG=.0	QG=.0	QG=.582D 06	QG=.203D 07	QG=.268D 07	QG=.612D 06	QG=.0	QG=.0
TVOL= 816.	TVOL= 1001.	TVOL= 1254.	TVOL= 1313.	TVOL= 1345.	TVOL= 1256.	TVOL= 999.	TVOL= 815.

REGION 80	REGION 56	REGION 34	REGION 18	REGION 19	REGION 8	REGION 21	REGION 40	REGION 64
QC=.0	QC=.0	QC=.109D 07	QC=.121D 07	QC=.113D 07	QC=.118D 07	QC=.910D 06	QC=.0	QC=.0
QS=.278D 07	QS=.769D 07	QS=.880D 07	QS=.914D 07	QS=.890D 07	QS=.902D 07	QS=.856D 07	QS=.761D 07	QS=.278D 07
QG=.0	QG=.0	QG=.149D 07	QG=.214D 07	QG=.163D 07	QG=.190D 07	QG=.993D 06	QG=.0	QG=.0
TVOL= 812.	TVOL= 1026.	TVOL= 1288.	TVOL= 1321.	TVOL= 1297.	TVOL= 1310.	TVOL= 1265.	TVOL= 1020.	TVOL= 812.

REGION 79	REGION 55	REGION 33	REGION 17	REGION 7	REGION 2	REGION 9	REGION 22	REGION 41	REGION 65	REGION 66
QC=.0	QC=.0	QC=.123D 07	QC=.114D 07	QC=.118D 07	QC=.121D 07	QC=.117D 07	QC=.926D 06	QC=.0	QC=.0	QC=.0
QS=.195D 07	QS=.742D 07	QS=.934D 07	QS=.893D 07	QS=.904D 07	QS=.915D 07	QS=.899D 07	QS=.858D 07	QS=.734D 07	QS=.195D 07	QS=.173D 07
QG=.0	QG=.0	QG=.258D 07	QG=.170D 07	QG=.193D 07	QG=.215D 07	QG=.183D 07	QG=.102D 07	QG=.0	QG=.0	QG=.0
TVOL= 817.	TVOL= 1006.	TVOL= 1340.	TVOL= 1301.	TVOL= 1311.	TVOL= 1322.	TVOL= 1306.	TVOL= 1266.	TVOL= 999.	TVOL= 816.	TVOL= 921.

REGION 54	REGION 32	REGION 16	REGION 6	REGION 1	REGION 3	REGION 10	REGION 23	REGION 42
QC=.0	QC=.792D 06	QC=.124D 07	QC=.106D 07	QC=.697D 06	QC=.121D 07	QC=.109D 07	QC=.620D 06	QC=.0
QS=.419D 07	QS=.935D 07	QS=.954D 07	QS=.877D 07	QS=.844D 07	QS=.919D 07	QS=.882D 07	QS=.907D 07	QS=.417D 07
QG=.0	QG=.117D 07	QG=.294D 07	QG=.136D 07	QG=.645D 06	QG=.222D 07	QG=.147D 07	QG=.674D 06	QG=.0
TVOL= 909.	TVOL= 1284.	TVOL= 1360.	TVOL= 1285.	TVOL= 1254.	TVOL= 1325.	TVOL= 1290.	TVOL= 1259.	TVOL= 906.

REGION 78	REGION 77	REGION 53	REGION 31	REGION 15	REGION 5	REGION 4	REGION 11	REGION 24	REGION 43	REGION 67
QC=.0	QC=.0	QC=.0	QC=.120D 07	QC=.123D 07	QC=.110D 07	QC=.120D 07	QC=.124D 07	QC=.111D 07	QC=.0	QC=.0
QS=.174D 07	QS=.195D 07	QS=.742D 07	QS=.911D 07	QS=.927D 07	QS=.884D 07	QS=.910D 07	QS=.949D 07	QS=.882D 07	QS=.737D 07	QS=.195D 07
QG=.0	QG=.0	QG=.0	QG=.212D 07	QG=.238D 07	QG=.152D 07	QG=.206D 07	QG=.283D 07	QG=.153D 07	QG=.0	QG=.0
TVOL= 822.	TVOL= 817.	TVOL= 1006.	TVOL= 1318.	TVOL= 1334.	TVOL= 1292.	TVOL= 1318.	TVOL= 1355.	TVOL= 1290.	TVOL= 1002.	TVOL= 916.

REGION 76	REGION 52	REGION 30	REGION 14	REGION 13	REGION 12	REGION 25	REGION 44	REGION 68
QC=.0	QC=.0	QC=.124D 07	QC=.122D 07	QC=.122D 07	QC=.119D 07	QC=.124D 07	QC=.0	QC=.0
QS=.279D 07	QS=.773D 07	QS=.974D 07	QS=.926D 07	QS=.925D 07	QS=.906D 07	QS=.950D 07	QS=.770D 07	QS=.278D 07
QG=.0	QG=.0	QG=.340D 07	QG=.236D 07	QG=.235D 07	QG=.196D 07	QG=.291D 07	QG=.0	QG=.0
TVOL= 813.	TVOL= 1030.	TVOL= 1380.	TVOL= 1332.	TVOL= 1332.	TVOL= 1313.	TVOL= 1356.	TVOL= 1027.	TVOL= 813.

Fig. 3.9.4.1. (Contd.)

*****FT.ST.VRIAN DBDA2 IMPLICIT 9-3-76*****

TIME= 0.25000 HOURS, TOTAL FLOW= 27275.6 LB/HR PRESSURE DROP(PSI)= 0.1312
 EXIT MEAN TEMPERATURE 1491.4269 UPPER PLEN. TEMP= 320.4201
 MAX SOLID TEMP= 1714.2577, MAX EXIT TEMP= 1503.6067

REGION 75 REGION 51 REGION 29 REGION 28 REGION 27 REGION 26 REGION 45 REGION 69
 QC=.0 QC=.0 QC=.794D 06 QC=.943D 06 QC=.122D 07 QC=.846D 06 QC=.0 QC=.0
 QS=.195D 07 QS=.744D 07 QS=.935D 07 QS=.860D 07 QS=.923D 07 QS=.954D 07 QS=.744D 07 QS=.195D 07
 QG=.0 QG=.0 QG=.119D 07 QG=.106D 07 QG=.236D 07 QG=.152D 07 QG=.0 QG=.0
 TVOL= 817. TVOL= 1007. TVOL= 1284. TVOL= 1269. TVOL= 1329. TVOL= 1301. TVOL= 1007. TVOL= 817.

REGION 50 REGION 49 REGION 48 REGION 47 REGION 46
 QC=.0 QC=.0 QC=.0 QC=.0 QC=.0
 QS=.419D 07 QS=.738D 07 QS=.767D 07 QS=.742D 07 QS=.420D 07
 QG=.0 QG=.0 QG=.0 QG=.0 QG=.0
 TVOL= 909. TVOL= 1002. TVOL= 1025. TVOL= 1006. TVOL= 910.

REGION 74 REGION 73 REGION 72 REGION 71 REGION 70
 QC=.0 QC=.0 QC=.0 QC=.0 QC=.0
 QS=.174D 07 QS=.195D 07 QS=.278D 07 QS=.195D 07 QS=.174D 07
 QG=.0 QG=.0 QG=.0 QG=.0 QG=.0
 TVOL= 822. TVOL= 816. TVOL= 812. TVOL= 817. TVOL= 822.

TOTAL HEAT REMOVED BY CUCLANT= 3.968530D 07 NET= 3.968530D 07
 TOTAL UP FLOW= 0.0
 TOTAL DOWN FLOW= 2.727557D 04
 ACTIVE CORE VOL.AVE TEMPERATURE 1400.20
 TOTAL HEAT RETAINED= 5.466088D 08
 BTU/HR CHANGE IN MODEL= 5.331189D 07
 UPPER REFLECTOR TEMPERATURES
 REFLECTOR MAX TEMP= 779.76 AT 52 MIN TEMP= 668.56 AT 11 AVE TEMP= 726.01
 LOWER REFLECTOR TEMPERATURES
 REFLECTOR MAX TEMP= 1470.42 AT 11 MIN TEMP= 829.24 AT 64 AVE TEMP= 1252.99
 SIDE REFLECTOR TEMPERATURES
 REFLECTOR MAX TEMP= 847.04 AT 580 MIN TEMP= 773.76 AT 66 AVE TEMP= *****
 ACTIVE CORE POWER (BTU/HR.)= 0.073122D 08
 POWER IN KW= 19608.0
 PERCENT OF FULL POWER= 2.32875

PRESSURE CHANGE ACROSS CORE -1.3120D-01 PSI

PRESSURE CHANGES FROM TOP TO BOTTOM OF CORE

	ORIFICE	ENTR (EXIT)	FRICT	ACCEL	DENHD	EXIT (ENTR)	NORM. FLOW
1	-6.0480D-02	-7.3954D-05	-6.5376D-02	-2.8814D-04	4.8488D-04	5.3773D-04	
2	-1.2112D-02	-2.1558D-04	-1.1984D-01	-8.6179D-04	4.7560D-04	1.3572D-03	
3	-1.1077D-02	-2.1755D-04	-1.2088D-01	-8.7058D-04	4.7481D-04	1.3703D-03	
4	-1.3534D-02	-2.1246D-04	-1.1841D-01	-8.4828D-04	4.7649D-04	1.3368D-03	
5	-2.5231D-02	-1.8226D-04	-1.0669D-01	-7.2170D-04	4.8139D-04	1.1425D-03	
6	-3.0236D-02	-1.6851D-04	-1.0176D-01	-6.6546D-04	4.8258D-04	1.1496D-03	
7	-1.5658D-02	-2.0771D-04	-1.1629D-01	-8.2777D-04	4.7777D-04	1.3058D-03	
8	-1.6376D-02	-2.0606D-04	-1.1562D-01	-8.2076D-04	4.7810D-04	1.2952D-03	
9	-1.7784D-02	-2.0245D-04	-1.1415D-01	-8.0551D-04	4.7831D-04	1.2718D-03	
10	-2.0639D-02	-1.7837D-04	-1.0527D-01	-7.0567D-04	4.8179D-04	1.1177D-03	
11	-5.0264D-03	-2.2480D-04	-1.2693D-01	-9.0642D-04	4.6832D-04	1.4211D-03	
12	-1.5167D-02	-2.0854D-04	-1.1678D-01	-8.3153D-04	4.7733D-04	1.3114D-03	
13	-9.4516D-03	-2.2038D-04	-1.2250D-01	-8.8339D-04	4.7346D-04	1.3892D-03	

Fig. 3.9.4.1. (Contd.)

14	-9.3551D-03	-2.2047D-04	-1.2260D-01	-8.8386D-04	4.7333D-04	1.3899D-03
15	-9.0993D-03	-2.2075D-04	-1.2286D-01	-8.8534D-04	4.7305D-04	1.3919D-03
16	-4.2963D-03	-2.2487D-04	-1.2766D-01	-9.0772D-04	4.6718D-04	1.4223D-03
17	-2.0568D-02	-1.9494D-04	-1.1136D-01	-7.7421D-04	4.7985D-04	1.2237D-03
18	-1.2201D-02	-2.1545D-04	-1.1975D-01	-8.6112D-04	4.7568D-04	1.3563D-03
19	-2.2218D-02	-1.9047D-04	-1.0971D-01	-7.5563D-04	4.8042D-04	1.1950D-03
20	-5.1789D-02	-1.1037D-04	-7.9943D-02	-4.2542D-04	4.8484D-04	5.8663D-04
21	-4.5756D-02	-1.2594D-04	-8.6224D-02	-4.9132D-04	4.8510D-04	9.1630D-04
22	-4.4187D-02	-1.3026D-04	-8.7803D-02	-5.0847D-04	4.8501D-04	9.4799D-04
23	-4.7031D-02	-1.2331D-04	-8.4707D-02	-4.7581D-04	4.8478D-04	6.5574D-04
24	-2.5079D-02	-1.8353D-04	-1.0684D-01	-7.2314D-04	4.8181D-04	1.1478D-03
25	-4.4969D-03	-2.2591D-04	-1.2746D-01	-9.0723D-04	4.6794D-04	1.4253D-03
26	-1.1982D-02	-2.2602D-04	-1.1979D-01	-8.8291D-04	4.7829D-04	1.2090D-03
27	-9.3968D-03	-2.2153D-04	-1.2256D-01	-8.8367D-04	4.7387D-04	1.3932D-03
28	-4.2359D-02	-1.3514D-04	-8.9643D-02	-5.2801D-04	4.8471D-04	9.8394D-04
29	-2.1036D-02	-2.0028D-04	-1.1073D-01	-7.7900D-04	4.8157D-04	1.0691D-03
30	-1.9001D-03	-2.2327D-04	-1.3005D-01	-9.0137D-04	4.6261D-04	1.4123D-03
31	-1.2581D-02	-2.1534D-04	-1.1937D-01	-8.5634D-04	4.7624D-04	1.3523D-03
32	-2.1315D-02	-1.9938D-04	-1.1045D-01	-7.7544D-04	4.8161D-04	1.0643D-03
33	-7.0893D-03	-2.2457D-04	-1.2487D-01	-8.9834D-04	4.7151D-04	1.4142D-03
34	-2.6238D-02	-1.8033D-04	-1.0568D-01	-7.1002D-04	4.8216D-04	1.1274D-03
35	-5.4299D-02	-1.0380D-04	-7.7430D-02	-3.9990D-04	4.8486D-04	5.5159D-04
36	-1.4108D-02	-2.1223D-04	-1.1784D-01	-8.4270D-04	4.7732D-04	1.3318D-03
37	-6.2482D-03	-2.2541D-04	-1.2571D-01	-9.0269D-04	4.7053D-04	1.4202D-03

THE CURRENT TIME STEP (DELTAT) = 7.50000000D-02

Fig. 3.9.4.1. (Contd.)

4. HEXEREI2 EVALUATION

In an effort to verify that all options in HEXEREI2 work as planned and to provide information on some of the areas where the analysis technique must be used with care, several studies were performed. The most important of these studies will be reported below. It should be noted that some current and planned code development is directed towards improvement of the inaccuracies found in the technique. The results presented below are intended to demonstrate the area and magnitude of these inaccuracies. As new options are added to HEXEREI2, this group of studies will be expanded and reported again to keep the user aware of improvements.

The areas of improvement include the coolant pressure drop, flow redistribution, CACS heat exchanger, CACS circulator and natural convection calculational techniques.

Both the transient explicit and implicit techniques were used in the following analysis and both are presented where significant differences occur.

4.1. Three-Dimensional Conduction

All of the following studies were performed on the model shown in Fig. 4.1. The material properties were uniform throughout the model. Both linear and temperature dependent thermal properties were used and are shown in Table 4.1. Conductive heat transfer was evaluated for radial conduction only in both steady-state and transient problems. The axial conduction was evaluated for the steady-state technique only. Figures 4.2 through 4.4 present the results for these analyses compared with HEATING5 (CCCM5) and when available an analytical solution for the problem.

Studies of the effect of mesh refinement have determined that 1) the finer the mesh the more accurate the solution is in hexagonal portion of the mesh and 2) an error is introduced by the interface between the regular hexagonal and cylindrical mesh that is independent of mesh size. Future code development will reduce this error as much as possible.

Table 4.1. Material Properties

LINEAR MATERIAL PROPERTIES		
Conductivity	—	10 Btu/hr-ft °F
Specific Heat	—	.45 Btu/lb _m °F
Density	—	10 lb _m /ft ³
NONLINEAR MATERIAL PROPERTIES		
Conductivity	—	100 e ^{(-.00005)T}
Specific Heat	—	45 e ^{(-.00005)T}
Density	—	100 e ^{(-.00005)T}

Fig. 4.1. Model for General Test Cases

Fig. 4.2. Comparison of Radial Temperature Distribution
Test Case Number 1 (Uniform Heat Generation, Constant
Properties, Constant Boundary Temperatures)

Fig. 4.3. Comparison of Radial Temperature Distribution
Test Case Number 2 (Uniform Heat Generation, Nonlinear
Properties, Constant Boundary Temperatures)

Fig. 4.4. Comparison of Radial Temperature Distribution for Transient Test Case with Constant Properties and Boundary Temperature and No Heat Generation

4.2. Boundary Heat Transfer

The very general surface to boundary heat transfer options in HEXEREI2 were partially evaluated by two analyses. A forced convective boundary condition was applied in turn to the circumference and then to the ends of the model cylinder. Figures 4.5 and 4.6 summarize these analyses and compare with analytical results.

4.3. Coupled Convection To Flowing Coolant

The coupled coolant-solid temperature calculational technique was evaluated for: a single channel; fixed coolant flow, pressure and composition; steady-state and transient problem. In this exercise, an attempt was made to simulate the same conditions that would exist during a depressurization accident from full flow and power. HEXEREI2's results were compared with CCCM5's for analogous models (Fig. 4.7). The steady-state results show good agreement for both the axial coolant temperature distribution (Fig. 4.8) and the volume average solid temperatures (Fig. 4.9). The volume average solid temperatures were necessary because CCCM5 must have at least two nodes in the radial direction and HEXEREI2's single node temperature represents the graphite volume average temperature. Similar results are presented for the transient analysis in Figs. 4.10 and 4.11.

Fig. 4.5. Steady-State Axial Temperature Distribution for Test Case with Boundary Condition Applied to Both Ends of Model

Fig. 4.6. Steady-State Radial Temperature Distribution for Test Case with Boundary Condition Applied to Outer Radius

Fig. 4.7. Analogous Models for HEXEREI and CCCM5

Fig. 4.8. Comparison of Axial Coolant Temperature Distribution.
Test Case Number 7 (Transient, Uniform and Constant Properties,
Heat Generation and Coolant Flow Rate)

Fig. 4.9. Comparison of Axial Solid Temperature Distribution.
Test Case Number 7 (Transient, Uniform and Constant Properties,
Heat Generation, and Coolant Flow Rate).
CCCM Points Are Volume Temperatures

Fig. 4.10. Comparison of Axial Coolant Temperature Distribution.
(Transient, Uniform and Constant Properties, Heat Generation
and Coolant Flow Rate)

Fig. 4.11. Comparison of Axial Solid Temperature Distribution.
 Test Case Number 7 (Transient, Uniform, and Constant Properties,
 Heat Generation, and Coolant Flow Rate).
 CCCM Points Are Volume Average Temperatures

5. DISCUSSION OF HEXEREI2 ANALYSES AS A USER AID

5.1. Sample Input for General Test Cases

The discussion in the previous section presented the results of HEXEREI2 for problems of a general nature. The problems demonstrate some of the capabilities of HEXEREI2 to analyze non HTGR-type problems. The HEXEREI2 input for test cases 1, 5, and 7 are included in Figs. 5.1.1 through 5.1.3 and are intended only as further clarification of the modeling technique used in the HEXEREI2 input data.

5.2. Fort St. Vrain Analysis

Extensive analysis has been performed on the Fort St. Vrain HTGR undergoing reactor trip accidents in response to NRC requests and to aid in debugging HEXEREI2. A discussion of the modeling techniques used may be of interest to future HEXEREI2 users in HTGR analysis. The reader should familiarize himself with input data format discussion in section 3.

These analyses were all performed on the same geometric model of the Fort. St. Vrain reactor consisting of a right circular cylinder with regions of different thermal properties or functions. These regions correspond to the upper, lower, and side reflectors, the active core and the core support blocks (Fig. 5.2.1). The outer surface of the right circular cylinder was assumed to be adiabatic or have convective or radiative heat transfer to a fixed boundary temperature.

Figure 5.2.2 shows a sample input deck to demonstrate the input of geometric model parameters. Card images that contain geometric model parameters are notated by an asterisk in column 80.

The material properties were modeled by a combination of simple data input and the more complex input technique of user-supplied subroutines. The card images in Fig. 5.2.2 that deal directly and indirectly are marked with an M in column 81 and an MI in columns 81 and 82, respectively. The indirect references include the analytic function cards calling for user-supplied subroutines (Cards A1). Figure 5.2.3 contains the FORTRAN listings of the user-supplied subroutines for specific heat (CPHEAT) used in the Fort St. Vrain analysis. Since HEXEREI2 is a lumped node type analysis technique, the fine structure of fuel rods and coolant holes are smeared or homogenized. In order to include the temperature variation of the specific heat capacity, the separate heat capacities for the fuel and moderator are combined in proportion to their mass in the node. The temperature for the temperature-dependent function is assumed to be the same for the fuel and moderator within a node.

It is possible to include the effect of anisotropic thermal conductivity with the use of a user-supplied subroutine (CONDTN) as shown in Fig. 5.2.4.

The region power peaking factors (RPPF) were taken from Ref. 24 and are presented in Fig. 5.2.5. The axial power distribution taken from Ref. 3 was input into HEXEREI2 as shown in Fig. 5.2.2. The RPPF is contained in entry 4 of the REG cards for the region (the first entry). The axial power distribution is modeled as a position-dependent heat generation where the variation in the z direction is detailed by the TB cards for the fourth tabular function. This tabular function number is input as a negative integer in the z-dependency (entry 7) of the heat generation card.

The time-dependency of the heat generation for these analyses was modeled in HEXEREI2 with a user-supplied subroutine technique and was taken from Ref. 25. Figure 5.2.6 lists the FORTRAN subroutine used. The user-supplied subroutine HEATGN modifies the passed value using a normalized time function. However, this technique could be modified in a manner to return any value the user desires.

TEST CASE NO. 1 FOR HEXEREI 7-28-75										CARD 2
120	1	2	2	1	2	1				CARD 3
3	2	2								CARD 4
	2			6						CARD 5
-2	500	1.0 E-8	1.9	.01	5	0.0	1.0			R1 6
1	1	.000001	13.850			0.0	20.8			R2 7
1	1									R1 8
2	2	13.85	18.85			0.0	20.8			R2 9
2	1		1							M1 10
1	CORE	10.0	10.0	0.45						M2 11
2	SIDE	10.0	10.0	0.45						G1 12
1	782.914									I1 13
1	1500.0									I2 14
2	200.0									B1 15
1	2	200.0								B2 16
										L1R 17
0.000001	13.85	18.85								N1R 18
-1	1									HXP 19
3.14526	1									L1T 20
0.0	3.14159									N1T 21
10										L1Z 22
0.0	20.8									N1Z 23
2										O1 24
0.5	1.0									S 25
1	1	2	8	20	38	62				

Fig. 5.1.1. Data for Test Case Number 1
(Fig. 4.2 summarizes the results of this model.)

TEST CASE NO. 5 FOR HEXEREI 8-26-75										CARD 2
120	1	2	2		2	1				CARD 3
3	2	2								CARD 4
	2			6						CARD 5
-1	500	1.0 E-8	1.9	.1	5	0.0	100.0			R1 6
1	1	.000001	13.850			0.0	20.8			R2 7
1										R1 8
2	2	13.85	18.85			0.0	20.8			R2 9
1			1							M1 10
1	CORE	10.0	10.0	0.45						M2 11
2	SIDE	10.0	10.0	0.45						I1 12
1	1500.0									I2 13
2	200.0									B1 14
1	2	200.0								B2 15
										L1R 16
0.000001	13.85	18.85								N1R 17
-1	1									HXP 18
3.14526	1									L1T 19
0.0	3.14159									N1T 20
10										L1Z 21
0.0	20.8									N1Z 22
2										O1 23
50.0	100.0									S 24
1	1	2	8	20	38	62				

Fig. 5.1.2. Data for Test Case Number 5
(Fig. 4.4 summarizes the results of this model.)

TEST CASE NO. 7 COOLANT CHECK CASE FOR HEXERE1 11-14-75										CARD 2	
100	1	2	2	2	1	2				5	CARD 3
3	2	2			3						CARD 4
	10				2						CARD 5
2	1000	1.0E-5	1.9	.01	2	0.0	1.0	27.8	R1	6	
1	1	.000001	13.850			0.0			R2	7	
1	1								R1	8	
2	2	13.85	18.85			0.0	20.8		R2	9	
2	1		1						M1	10	
1	CORE	10.0	10.0	0.45					M2	11	
2	SIDE	10.0	10.0	0.45					G1	12	
1	533144.	-3							I1	13	
1	400.0								I2	14	
2	200.0								L1R	15	
0.000001	13.85	18.85							N1R	16	
-1	1								HXP	17	
3.14526	1								L1T	18	
0.0	3.14159								N1T	19	
10									L1Z	20	
0.0	20.8								N1Z	21	
5									TB1	22	
1	3								TB2	23	
0.0	1.0	1.0E-4	.05	2.0	.05				TB1	24	
2	3								TB2	25	
0.0	1.0	.01	.66	2.0	.66				TB1	26	
3	4								TB2	27	
0.0	1.0	.0001	.01	0.01	.01	2.0	.01		Q1	28	
0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8		Q2	29	
0.9	10.0								S	30	
2	1	91							HTGR31		
61	699.	600.0	4.0	82988.0	.0883333	.08833			HTFN32		
	-1	-2							REG	33	
1	0	250	1.0	1.0	1.0	82988.0			REG	34	
2	0	250	1.0	1.0	1.0	82988.0			REG	35	
3	0	250	1.0	1.0	1.0	82988.0			REG	36	
4	0	250	1.0	1.0	1.0	82988.0			REG	37	
5	0	250	1.0	1.0	1.0	82988.0			REG	38	
6	0	250	1.0	1.0	1.0	82988.0			REG	39	
7	0	250	1.0	1.0	1.0	82988.0			REG	40	
8	0	250	1.0	1.0	1.0	82988.0			REG	41	
9	0	250	1.0	1.0	1.0	82988.0			REG	42	
10	0	250	1.0	1.0	1.0	82988.0			REG	43	
11	0	250	1.0	1.0	1.0	82988.0			REG	44	
12	0	250	1.0	1.0	1.0	82988.0			REG	45	
13	0	250	1.0	1.0	1.0	82988.0			REG	46	
14	0	250	1.0	1.0	1.0	82988.0			REG	47	
15	0	250	1.0	1.0	1.0	82988.0			REG	48	
16	0	250	1.0	1.0	1.0	82988.0			REG	49	
17	0	250	1.0	1.0	1.0	82988.0			REG	50	
18	0	250	1.0	1.0	1.0	82988.0			REG	51	
19	0	250	1.0	1.0	1.0	82988.0			REG	52	
20	0	250	1.0	1.0	1.0	82988.0			REG	53	
21	0	250	1.0	1.0	1.0	82988.0			REG	54	
22	0	250	1.0	1.0	1.0	82988.0			REG	55	
23	0	250	1.0	1.0	1.0	82988.0			REG	56	
24	0	250	1.0	1.0	1.0	82988.0			REG	57	
25	0	250	1.0	1.0	1.0	82988.0			REG	58	
26	0	250	1.0	1.0	1.0	82988.0			REG	59	
27	0	250	1.0	1.0	1.0	82988.0			REG	60	
28	0	250	1.0	1.0	1.0	82988.0			REG	61	
29	0	250	1.0	1.0	1.0	82988.0			REG	62	
30	0	250	1.0	1.0	1.0	82988.0			REG	63	
31	0	250	1.0	1.0	1.0	82988.0			REG	64	
32	0	250	1.0	1.0	1.0	82988.0			REG	65	
33	0	250	1.0	1.0	1.0	82988.0			REG	66	
34	0	250	1.0	1.0	1.0	82988.0			REG	67	
35	0	250	1.0	1.0	1.0	82988.0			REG	68	
36	0	250	1.0	1.0	1.0	82988.0			REG	69	
37	0	250	1.0	1.0	1.0	82988.0			REG	70	
38	0	250	1.0	1.0	1.0	82988.0			REG	71	
39	0	250	1.0	1.0	1.0	82988.0			REG	72	
40	0	250	1.0	1.0	1.0	82988.0			REG	73	
41	0	250	1.0	1.0	1.0	82988.0			REG	74	
42	0	250	1.0	1.0	1.0	82988.0			REG	75	
43	0	250	1.0	1.0	1.0	82988.0			REG	76	
44	0	250	1.0	1.0	1.0	82988.0			REG	77	
45	0	250	1.0	1.0	1.0	82988.0			REG	78	
46	0	250	1.0	1.0	1.0	82988.0			REG	79	
47	0	250	1.0	1.0	1.0	82988.0			REG	80	
48	0	250	1.0	1.0	1.0	82988.0			REG	81	
49	0	250	1.0	1.0	1.0	82988.0			REG	82	
50	0	250	1.0	1.0	1.0	82988.0			REG	83	
51	0	250	1.0	1.0	1.0	82988.0			REG	84	
52	0	250	1.0	1.0	1.0	82988.0			REG	85	
53	0	250	1.0	1.0	1.0	82988.0			REG	86	
54	0	250	1.0	1.0	1.0	82988.0			REG	87	
55	0	250	1.0	1.0	1.0	82988.0			REG	88	
56	0	250	1.0	1.0	1.0	82988.0			REG	89	
57	0	250	1.0	1.0	1.0	82988.0			REG	90	
58	0	250	1.0	1.0	1.0	82988.0			REG	91	
59	0	250	1.0	1.0	1.0	82988.0			REG	92	
60	0	250	1.0	1.0	1.0	82988.0			REG	93	
61	0	250	1.0	1.0	1.0	82988.0					

Fig. 5.1.3. Data for Test Case Number 7
(Figs. 4.8 through 4.11 summarize the results of this model.)

Fig. 5.2.1. Geometric Model for HEXEREI2 Analysis of Fort St. Vrain HTGR

COLS.	1	2	3	4	5	6	7	8
9	8	7	6	5	4	3	2	0
FT. ST. VRAIN FOR EASE CASE NO. 3-IMPLICIT- 40% POWER (4-30-76)								
840	1	4	2	1	2	1		CARD 2
3	2	4	2	5			5	CARD 3
1	0					0		CARD 4
2	500	1.00-5	1.9		0	0.0	10.0	CARD 5
1	1	.000001	9.810			3.9	19.5	R1 6*M
1	1							R2 7*
2	2	9.810	13.67			3.9	19.5	R1 8*M
2			1					R2 9*
3	2	0.0	13.67			0.0	3.9	R1 10*M
2			1					R2 11*
4	2	0.0	13.67			19.5	25.34	R1 12*M
2			1					R2 13*
1	CORE	15.0	1.00	1.00				2 M1 14 M
2	REFLECT	15.0	107.0	1.00				2 M2 15 M
1	2.4217C5	1				-4		G1 16
1	400.0							I1 17
2	200.0							I2 18
1	1	1.0	-3					B1 19
11.37								B2 20
0.000001	9.810	13.67						L1R 21*
-1	1							N1R 22*
3.13333	1							HXP 23*
0.0	3.14159							L1T 24*
10								N1T 25*
0.0	3.9	19.5	25.34					L1Z 26*
1	6	1	1					N1Z 27*
1								A1 28
2								A1 29 M1
1	19							TB1 30
0.0	569.	3.220-3	569.00	8.220-3	568.	.0182	567.	TB2 31
2.820-2	565.00	3.820-2	564.	.0482	563.	5.820-2	561.00	TB2 32
6.820-2	559.	.0882	556.	1.080-1	552.00	1.260-1	549.	TB2 33
.1460	545.	1.700-1	540.00	1.980-1	536.	.2210	533.	TB2 34
2.460-1	529.00	2.500-1	529.	.25001	529.			TB2 35
2	20							TB1 36
0.0	2.7539704	4.20-4	1.929204	3.220-3	1.933704	8.220-3	1.953004	TB2 37
1.820-2	1.989804	2.820-2	2.026604	3.820-2	2.045904	4.820-2	2.102004	TB2 38
5.820-2	2.140604	6.820-2	2.177404	8.820-2	2.252804	1.080-1	2.330204	TB2 39
1.260-1	2.330204	1.480-1	2.316604	1.700-1	2.303104	1.980-1	2.285704	TB2 40
2.210-1	2.272104	2.460-1	2.256704	2.500-1	2.256704	2.50010-1	0.0000	TB2 41
3	20							TB1 42
0.0	600.	.00042	600.	.00322	599.	.00822	597.	TB2 43
0.0182	592.	0.0282	588.	0.0382	584.	0.0482	579.	TB2 44
0.0582	575.	0.0682	571.	0.0882	562.	0.1080	554.	TB2 45
0.1260	546.	0.148	537.	0.1700	527.	0.1980	515.	TB2 46
0.2210	506.	0.246	495.	0.2500	495.	.25001	495.	TB2 47
4	15							TB1 48
0.0	0.0	3.8999	0.0	3.90	0.8205	6.499	0.8205	TB2 49
6.5	1.1498	9.0999	1.1498	9.1	1.3269	11.699	1.3269	TB2 50
11.7	1.0787	14.299	1.0787	14.3	0.9096	16.699	0.9096	TB2 51
16.700	0.7145	19.499	0.7145	19.500	0.0			TB2 52
5	22							TB1 53
0.0	.400	1.00-5	.362	.125	.35	.166666	.345	TB2 54
.222222	.340	.27777	.335	.55555	.320	1.0	.300	TB2 55
1.52777	.280	2.5	.26	4.1666	.24	6.9444	.22	TB2 56
11.1111	.2	17.777	.18	33.333	.16	55.555	.14	TB2 57
83.3333	.127	111.111	.12	138.888	.114	194.444	.108	TB2 58
277.7777	.102	416.666	.102					TB2 59

Fig. 5.2.2. Data for HEXEREI2 Analysis of Fort St. Vrain HTGR

COLS.	1	2	3	4	5	6	7	8
9	8	7	6	5	4	3	2	0
37	1.00	1.00	4.0	.950	.052087	.04167		HTGR60
	-1	-3		-2				HTFN61
1	41	664	0.33833	1.0	1.0	69000.0		REG 62
2	41	664	1.1274	1.0	1.0	69000.0		REG 63
3	41	664	1.1660	1.0	1.0	69000.0		REG 64
4	41	664	1.0785	1.0	1.0	69000.0		REG 65
5	41	664	0.7964	1.0	1.0	69000.0		REG 66
6	41	664	0.7131	1.0	1.0	69000.0		REG 67
7	41	664	1.0140	1.0	1.0	69000.0		REG 68
8	41	664	0.9952	1.0	1.0	69000.0		REG 69
9	41	664	0.9567	1.0	1.0	69000.0		REG 70
10	41	664	0.7714	1.0	1.0	69000.0		REG 71
11	41	664	1.4835	1.0	1.0	69000.0		REG 72
12	41	664	1.0275	1.0	1.0	69000.0		REG 73
13	41	664	1.2336	1.0	1.0	69000.0		REG 74
14	41	664	1.2378	1.0	1.0	69000.0		REG 75
15	41	664	1.2492	1.0	1.0	69000.0		REG 76
16	41	664	1.5397	1.0	1.0	69000.0		REG 77
17	41	664	0.8901	1.0	1.0	69000.0		REG 78
18	41	664	1.1243	1.0	1.0	69000.0		REG 79
19	41	664	0.8547	1.0	1.0	69000.0		REG 80
20	29	460	0.4495	0.714286	1.0	69000.0		REG 81
21	41	664	0.5205	1.0	1.0	69000.0		REG 82
22	41	664	0.5372	1.0	1.0	69000.0		REG 83
23	29	460	0.4949	0.714286	1.0	69000.0		REG 84
24	41	664	0.8016	1.0	1.0	69000.0		REG 85
25	41	664	1.5261	1.0	1.0	69000.0		REG 86
26	29	460	1.1140	0.714286	1.0	69000.0		REG 87
27	41	664	1.2388	1.0	1.0	69000.0		REG 88
28	41	664	0.5569	1.0	1.0	69000.0		REG 89
29	29	460	0.8656	0.714286	1.0	69000.0		REG 90
30	41	664	1.7843	1.0	1.0	69000.0		REG 91
31	41	664	1.1129	1.0	1.0	69000.0		REG 92
32	29	460	0.8595	0.714286	1.0	69000.0		REG 93
33	41	664	1.3544	1.0	1.0	69000.0		REG 94
34	41	664	0.7808	1.0	1.0	69000.0		REG 95
35	29	460	0.4272	0.714286	1.0	69000.0		REG 96
36	41	664	1.0629	1.0	1.0	69000.0		REG 97
37	41	664	1.4043	1.0	1.0	69000.0		REG 98
0.629166	1.75	22.92	24.17					CSB 99
1.14583	.075	.96875	0.623295	8.0764	1.8403	2.403		CAT100
1.00-3	1.00-3	1.00-3	.50	0.0				IP 101
.001	1.00		.25					TP 102
.001	1.00		.50					TP2103
.001	1.2		10.					TP3104

Fig. 5.2.2. (Contd.)

```

SUBROUTINE CPHEAT(RVALUE,N,TH,Z,TM,TSN,VALUE,NUMBER,N)
IMPLICIT REAL*8 (A-H,C-Z)
C      FOR CALCULATIONS INVOLVING COOLANT FLOW THE ABSOLUTE
C      TEMPERATURE FLAG IS SET ON(IRAD).
COMMON /ITCCHD/ TABS(2) , JABS , IRAD ,TDCDHC
COMMON /ERROR/ IERROR
C      THE MODIFICATIONS TO THE NODES CPHEAT DUE TO THE
C      HOMOGENIZATION OF THE FUEL AND MODERATOR REQUIRE
C      THE INCLUSION OF THE FOLLOWING COMMON BLOCK.
COMMON/HEXCOR/AZ(200),C(6,200),KC(6,200),NCCOM(6,200),NLEV,NPG,NPI
INTEGER*2 KC,NCLDM
C
C      COMMON BLOCK KOOLNT CONTAINS PARAMETERS USED FOR COOLANT
C      CALCULATIONS FOR THE HTGR. IT MUST APPEAR IN SUBROUTINES
C      MINPT1,HTHRMP,HCALGT,EXPAN,EXPAME,AND HTRAND.
COMMON /KOOLNT/ JPPF(121),PCACFE(121),DRAREA(121),G(121),
.      FLOARE(121),TEXT(121),DIAMTR,DIAMTS,NCLCHS(121),
.      NCLCH(121),COOLNT,ORISZE
COMMON /UPERPL/ PCON,TCON,COOLMW,GTOT,NPCTM,NTCTM,NCMWTM,NGTTM
LOGICAL COOLNT,ORISZE
LOGICAL MSGE /.FALSE./
DATA TADJ/0.000/
C
IF(MSGE)GO TO 10
MSGE=.TRUE.
NUM=1
RHOM = 107.000
IF(IRAD.LT.2) TAJ = TABS(JABS)
WRITE(6,2000)
2000 FORMAT('THE SPECIFIC HEAT FOR THE MATERIALS DENOTED BELOW WILL BE
1 EVALUATED ACCORDING TO THE FOLLOWING ALGORITHMS.'/'NUMBER',29X,
2ALGORITM')
WRITE(6,2005)NUM,RHOM
2005 FORMAT(' ',IS,6X,'(RHO*CP)FUEL = 3.38 + 2.74E-2*T - 4.88E-6*T**2,
.      - 0.146E-6/T**2*/T12,'(CP)MODERATOR= -0.103 + 6.472E-4*T,
.      - 2.667E-7*T**2 + 3.97E-11*T**3*/T12,'MODERATOR DENSITY=',
.      F10.4)
WRITE(6,2011)
2011 FORMAT(' ',T12,' THE SPECIFIC HEAT OF THE NODE IS A FUNCTION',
.      OF THE PCACFE OF THE REGION SUCH THAT*/T12,' CP*RHO =',
.      (RHO*CP*VOLF(FUEL) + RHO*CP*VOLF(MODERATOR))*PCACFE +',
.      (1-PCACFE)*RHO*CP*VOLF(MODERATOR)*/T12,' WHERE VOLF =',
.      VOLUME FRACTION OF THAT MATERIAL')
NUM=NUM+1
WRITE(6,2010)NUM
2010 FORMAT(' ',IS,6X,'CP = -0.103 + 6.472E-4*T - 2.667E-7*T**2 + 3.97E
1-11*T**3')
C      CONVERT THE TEMPERATURES TO ABSOLUTE.
10 TSNABS = TSN + TADJ
C      IF IRAD=2 THEN TADJ=0.0
TSNSQ=TSNABS*TSNABS
TSNCU=TSNSQ*TSNABS
IF(NUMBER.GT.1)GO TO 310
C      ACTIVE CORE FUEL RHO*CP.
K = (N-1)/NLEV + 1
NH = N - (K-1)*NLEV
PCDUM = PCACFE(NH)
IF(PCDUM.LT.0.000) PCCLM = 0.000
IF(PCDUM.GT.1.000) PCDUM = 1.000
RVFUEL = 3.3800 + 0.027400*TSNABS - 4.0800*TSNSQ - 0.14606/TSNSQ
C      MODERATOR CP.
RVMOD = -0.10300 + 6.4720-4*TSNABS - 2.6670-7*TSNSQ + 3.970-11*
1 TSNCU
IF(RVMOD.GE.0.000) GO TO 1234
WRITE(6,10023) RVMOD,TSN,TSNABS,TSNSQ,TSNCU
10023 FORMAT(' MODERATOR SPECIFIC HEAT IS LESS THAN ZERO'*/T12,'CP=',
.      G20.10,'TSN=',G13.6,'TSNABS=',G13.6,'TSNSQ=',G13.6,
.      'TSNCU=',G13.6)
IF(IERROR.GT.0) WRITE(IERROR,10023) RVMOD,TSN,TSNABS,TSNSQ,TSNCU
RVMOD = 0.000
1234 CONTINUE
C      ACTIVE CORE HOMOGENIZED MODEL RHO*CP
RVALUE = (RVFUEL*0.24084 + RVMOD*RHOM*0.75916)*PCDUM
.      + (1.000-PCDUM)*RVMOD*RHOM
IF(RVALUE.GE.0.000) GO TO 1235
WRITE(6,10024) RVALUE,N,PCACFE(NH),NH,RVFUEL,RVMOD,PCDUM
10024 FORMAT(' RVALUE IS LESS THAN ZERO. RVALUE=',G20.10,' N=',IS,
.      PCACFE(NH)='G13.6,' NH='IS,' RVFUEL='G13.6,' RVMOD='G13.6,'
.      PCDUM='G13.6)
IF(IERROR.GT.0) WRITE(IERROR,10024) RVALUE,N,PCACFE(NH),NH,RVFUEL,
.      RVMOD,PCDUM
RVALUE = 0.000
1235 RETURN
310 CONTINUE
RVALUE = -0.10300 + 6.4720-4*TSNABS - 2.6670-7*TSNSQ + 3.970-11*
1 TSNCU
RETURN
END

```

Fig. 5.2.3. FORTRAN Listing of User-Supplied Subroutine CPHEAT

```

SUBROUTINE CONDTN(RVALUE,R,TH,Z,TIM,TSN,VALUE,NUMBER,N)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /THBSBC/ NBDTP,NDIR
COMMON/HEXCOR/AZ(200),C(6,200),KC(6,200),NCCOM(6,200),NLEV,NPO,NPI
INTEGER*2 KC,NCCOM

C
C      COMMON BLOCK KOOLNT CONTAINS PARAMETERS USED FOR COOLANT
C      CALCULATIONS FOR THE HTGR. IT MUST APPEAR IN SUBROUTINES
C      HINPT1,HTHRMP,HCALGT,EXFAM,EXPAME,AND HTRANO.
COMMON /KOOLNT/ RPPF(121),PCACFE(121),ORAREA(121),G(121),
.      FLOARE(121),TEXT(121),DIAMTR,DIAMTS,NCLCHS(121),
.      NCLCH(121),COOLNT,ORSIZE
COMMON /UPERPL/ PCON,TCON,COOLMW,GTOT,NPCTM,NTCTM,NCMTM,NGTMM
LOGICAL COOLNT,ORSIZE
DIMENSION AXRAT(2)
LOGICAL MSGE /.FALSE./
IF(MSGE)GO TO 10
MSGE=.TRUE.
AXRAT(1) = 3.61/2.81
AXRAT(2) = 1.0
NUM=1
WRITE(6,2000)
2000 FORMAT('0THE CONDUCTIVITY FOR THE MATERIALS DENOTED BELOW WILL BE
1 EVALUATED ACCORDING TO THE FOLLOWING ALGORITHMS.'/'0NUMBER',29X,'
2ALGORITHM')
WRITE(6,2005)NUM,AXRAT(NUM)
2005 FORMAT(' ',15.6X,'K(Z) = K',F10.5)
WRITE(6,2011)
2011 FORMAT(' ',T12,' THE CONDUCTIVITY OF THE NODE IS A FUNCTION',
.      ' OF THE PCACFE OF THE REGION SUCH THAT'/T12,' K(Z) =',
.      'K(Z)*PCACFE + (1-PCACFE)*K'/T12,' WHERE K = THE ',
.      'ISOTROPIC CONDUCTIVITY OF THE REFLECTOR GRAPHITE')

NUM=NUM+1
WRITE(6,2005)NUM,AXRAT(NUM)
10 IF(NDIR.GT.1)GO TO 310
C      ACTIVE CORE CONDUCTIVITY
K = (N-1)/NLEV + 1
NH = N - (K-1)*NLEV
PCDUM = PCACFE(NH)
IF(PCDUM.LT.0.000) PCDUM = 0.000
IF(PCDUM.GT.1.000) PCDUM = 1.000
C      ACTIVE CORE HOMOGENIZED MODEL CONDUCTIVITY
RVALUE = VALUE*AXRAT(NUM)*PCDUM + (1.000-PCDUM)*VALUE
310 CONTINUE
RVALUE = VALUE
RETURN
END

```

Fig. 5.2.4. FORTRAN Listing of User-Supplied Subroutine CONDTN Demonstrating Anisotropic Conductivity Modeling

Fig. 5.2.5. Region Power Peaking Factors


```

      SUBROUTINE HEATGN(RVALUE,R,TH,Z,TIM,TSN,VALUE,NUMBER,N)
C ***** NOTE *****
C      THIS ROUTINE HAS BEEN MODIFIED TO CALCULATE THE TIME-DEPENDENT
C      POWER FUNCTION ACCORDING TO EQ (5.10) OF GULF GA-A12504.
C      THIS ROUTINE IS A MODIFIED VERSION TO USE ON THE 40% POWER
C      FEB. 76 AD HOC STUDIES.
C      THIS ROUTINE REQUIRES A TABLE NO. 5 TO PROPERLY ADJUST THE
C      THE AFTER HEAT POWER.
C *****
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL MSGE / .FALSE. /
      IF (MSGE) GO TO 10
      F1      = 1.0
      DUM      = 1.0
      NTBL     = -5
      MSGE = .TRUE.
      WRITE(6,1000)
1000  FORMAT('***** NOTE -- THE NORMALIZED DECAY FUNCTION ASSOCIATED WI
      1TH THE POWER WILL BE EVALUATED AS'/
      2 'OF(T) = 0.128*( T*3600.0+0.0003796)**(-0.261)')
10  CONTINUE
      RVALUE=0.128D0*(TIM*3600.0D0+0.0003796D0)**(-0.261D0)
      CALL TABLE(NTBL,DUM,TIM,F1,N,NUMBER)
      RVALUE = F1/0.40D0*RVALUE
      RETURN
      END

```

Fig. 5.2.6. FORTRAN Listing of User-Supplied Subroutine HEATGN

The coolant channel model used was also input with the cards in Fig. 5.2.2. The Fort St. Vrain reactor has two general sizes of coolant channels. The card labeled 'HTGR' contains the diameters (last two entries) in the consistent units (feet) of both sizes of channels. The cards labeled 'REG' contain the number of each size present in each nodal column (the second and third entries). The information necessary to describe the orifice and upper reflector plenum is contained in the card labeled 'CAT'. This data includes the orifice maximum flow area (also the orifice cylinder area); the orifice minimum flow area; the orifice cylinder length and hydraulic diameter; the flow areas of the upper reflector plenum, first and second expansions in the lower reflector/core support blocks. The expansions from coolant channel to collector and then to core support block plenum are described on the card labeled 'CSB' and include the flow areas, diameters, and axial location of each expansion.

It was noted above that the outer boundaries of the model cylinder could be adiabatic or transfer heat by convection and/or radiation to a fixed boundary temperature. In the reactor, these surfaces transfer heat to the PCRV thermal barrier cover plates or the PCRV liner if the thermal barrier insulation has slumped due to excessive temperatures. These conditions do not represent a fixed temperature boundary. HEXEREI2 does not, at present, have the ability to determine the temperature distribution in the thermal barrier insulation or the PCRV, and so the approximation of the PCRV liner or thermal barrier surface as a fixed temperature was the only method available to model the heat loss to the PCRV liner cooling system.

The addition of an analysis of the steam generator and main circulator requires the inclusion of some optional cards shown in Fig. 5.2.2. The type of analysis (closed or open) is indicated by the sign of the first entry on the card labeled 'HTFN' in Fig. 5.2.2. The heat exchanger model parameters are included in the code and do not require user interaction.

5.3. Time Step Size Consideration

As mentioned above, the user must assure himself that the time step size used in his analysis is the optimum for accuracy within the computer cost and system constraints. Executing the analysis with a spectrum of time step sizes is the only proven method of assuring an accurate analysis. The user must be very cautious whenever time- or temperature-dependent parameters vary by large amounts for small changes in time or temperature.

ACKNOWLEDGMENTS

The authors would like to acknowledge all of the help and advice that Dr. John P. Sanders* offered during the development of the HEXEREI2 code. We would also like to thank Oliver Russ for his work on one of the output subroutines.

Marie Cuthbert and Shirley Boles deserve thanks for learning the techniques necessary to input this report into the computer so that it could be produced in its present format.

*Program Manager of HTGR Safety Studies Project in ORNL's Engineering Technology Division.

REFERENCES

1. H. W. Chi and G. J. Malek, *Description of the Reactor Emergency Cooling Analysis Code, RECA*, GA-10273 (Aug. 19, 1970).
2. W. D. Turner, *HEATING5 - An IBM 360 Heat Conduction Program*, Oak Ridge National Laboratory, Oak Ridge, Tennessee, March, 1977 (ORNL/CSD/TM-15).
3. Public Service Company of Colorado, *Fort St. Vrain Nuclear Generating Station, Final Safety Analysis Report*.
4. Philadelphia Electric Company, *Fulton Generating Station, Units 1 and 2, Preliminary Safety Analysis Report (PSAR)*.
5. W. D. Turner, Letter to Richard Lobel, Division of Technical Review, USNRC, Subject: Transmission of Coupled-Conduction Convection - Model (CCCM) Computer Code, Dated September 23, 1975.
6. J. P. Sanders *et al.*, *Evaluation of Thermal Response in Fort St. Vrain Reactor Primary System to a Design Basis Depressurization Accident Followed by Cooling with Two Pelton Wheel Drives at 7000 rpm*, Oak Ridge National Laboratory, Oak Ridge, Tennessee, December, 1975 (ORNL-TM-5140).
7. D. W. Savery, *TAP - A FORTRAN IV Program for the Transient Analysis of the HTGR Power Plant Performance*, GAMD-7248 (1967).
8. R. D. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problems*, Interscience Publishers, New York, New York, 1967.
9. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1962.
10. S. K. Hellman, G. Habetler, and H. Babrov, *Use of Numerical Analysis in the Transient Solution of Two-Dimensional Heat-Transfer Problem with Natural and Forced Convection*, Trans. Am. Soc. Mech. Engrs., 78, pp. 1155-61, 1956.
11. S. Levy, *Use of "Explicit Method" in Heat Transfer Calculations with an Arbitrary Time Step*, General Electric, Schenectady, New York, August, 1968 (GE Report No. 68-C-282).
12. B. A. Carré, *The Determination of the Optimum Accelerating Factor for Successive Over-Relaxation*, Comp. J., Vol. 4, pp. 73-78, 1961.
13. J. P. Sanders, "Physical and Thermodynamic Properties of Helium to be Used in Performance Evaluation of Gas-Cooled Reactors," Oak Ridge National Laboratory, Oak Ridge, Tennessee, letter dated November, 1972.
14. W. M. Rohsenow and J. P. Hartnett, *Handbook of Heat Transfer*, McGraw-Hill, 1973.
15. W. M. Kays, *Loss Coefficients for Abrupt Changes in Flow Cross Section with Low Reynolds Number Flow in Single and Multiple Tube Systems*, Transactions of ASME Paper No. 50-S-7, November 1950.
16. D. M. Bender, Goodyear Atomic Corporation, Letter to G. C. Bramblett Referenced FSVA:98:76 dated May 27, 1976.
17. J. R. Tallackson, *CACHE - An Extended Basic Program Which Computes the Performance of Shell and Tube Heat Exchangers*, Oak Ridge National Laboratory, Oak Ridge, Tennessee, March, 1976 (ORNL/TM-4952).

18. E. D. Grimison, *Correlation and Utilization of New Data on Flow Resistance and Heat Transfer for Cross Flow of Gases Over Tube Banks*, Trans. ASME, 59, pp. 583-593 (October, 1957).
19. M. Jakob, *Heat Transfer and Flow Resistance in Cross Flow Over Tube Banks*, Trans. ASME, 60, pp. 384-386 (1938).
20. R. F. Walker letter to R. A. Clark containing responses to question 14.11-4., letter dated February 28, 1975.
21. R. L. Reid, J. P. Sanders, *INGRES: A Computer Code for the Rate of Air Ingress into an HTGR Following a Design-Basis Depressurization Accident*, Oak Ridge National Laboratory, Oak Ridge, Tennessee, December 1975 (ORNL/TM-5144).
22. J. R. Tallackson, *The Thermal Transport Properties of Helium, Helium-Air Mixtures, Water and Tubing Steel Used in the CACHE Program to Compute HTGR Auxiliary Heat Exchanger Performance*, Oak Ridge National Laboratory, Oak Ridge, Tennessee, February, 1976 (ORNL/TM-4931).
23. J. H. Perry, *Chemical Engineers' Handbook*, McGraw-Hill Book Co., New York.
24. R. F. Walker, Letter to Robert A. Clark, Docket No. 50-267 (February 28, 1975).
25. R. F. Walker, Public Service Company of Colorado, Letter to Richard P. Denise, A/D Advanced Reactor Concepts, USNRC, Docket No. 50-267, Attachment 2: "Limiting Power Level Prior to Availability of Interim Alternate Cooling Method," December 30, 1975.

APPENDIX A

CONTROL CARDS

A version of HEXEREI2, which can be used to solve problems containing up to 765 nodes, is stored on disks at the ORGDP computing centers. This appendix discusses the Job Control Language (JCL) which is required to run this version of HEXEREI2 along with modifications to the JCL which are necessary for some of the optional uses of the code. This information will allow the user to solve most of his problems on the code. For additional information concerning JCL, the user is referred to the Oak Ridge Programmer's Notebook.

The control cards, which are necessary to use this version of HEXEREI at ORGDP are presented in Fig. A-1. The first and second control cards are called the JOB statement and CLASS statement, respectively. The job name has a format of UIDXX where UID is the userid assigned to the user by the Computer Sciences Division (CSD) and the characters XX are any characters which make the job name unique. The accounting field (cccc) contains a four- or five-digit CSD charge number. The programmer-name field contains information necessary to route the job output back to the user. Since the default on core size will be exceeded for this version of HEXEREI2, a CLASS card must be included. See the Oak Ridge Programmer's Notebook for additional information concerning the JOB and CLASS statements.

If a request to save the final temperature distribution is made by inserting a unit number in the seventh entry on Card 4 (see section 3.6.4(f)), then a DD card describing the output unit must appear before the

```
//GO.FT05F001 DD *
```

card. The following card

```
//GO.FT07F001 DD SYSOUT=B
```

would be a typical DD card for punching the final temperature distribution on cards on unit 7. The integer 7 would be entered in entry 7 of Card 4 to define the unit number. If a number other than 7 is specified, then the DCB parameters must be specified on the DD card.

```
//jobname JOB (cccc), 'programmer-name'
//*CLASS REGION=500K, CPUxx=yyyz, LINES=///
// EXEC FORTHLG, REGION.GO=500K, PARM.GO='EU=-1'
//LKED.HEXEREI2 DD DISP=SHR, DSN=A.GEGHD845.HEXEREI2
//LKED.SYSIN DD *
  INCLUDE HEXEREI2
/*
//GO.FT05F001 DD *
(Insert Data Deck Here)
/*
//
```

Fig. A-1. Format of JCL to Run HEXEREI2 at ORGDP

If one wishes to save the output temperature distributions in order to generate various plots with a plotting package, then a DD card describing the output data set must be inserted just before the

```
//GO.FT05F001 DD *
```

card. If a data set involving 450 nodes and two boundary conditions is to be saved on tape, then a typical DD card might be

```
//GO.FT08F001 DD UNIT=TAPE9,DISP=OLD,VOL=SER=TTTTTT,LABEL=(,SL),  
// DCB=(RECFM=VBS,LRECL=3620,BLKSIZE=3624)
```

where 8 is the unit number which must be inserted in entry 3 of Card 3. TTTTTT is the tape reel number.

Two trivial modifications in HEXEREI2 will allow the code to solve problems containing up to a maximum of 7000 nodes. The main program for HEXEREI2 is shown in Fig. A-2. Let N be the maximum number of nodes which are desired for this case. The variable N must be initialized in the INTEGER statement, and the variable core must be dimensioned 25 times N. The amount of core in K bytes required by HEXEREI2 is given roughly by the equation

$$M = (314 + 0.2N) . \quad (A-1)$$

Thus, the code can be easily modified to fit into a specified amount of core or to solve a problem with a specified number of nodes. The format of the JCL which is required to run HEXEREI2 with the main program modified is presented in Fig. A-3. The modified MAIN routine is inserted in the location indicated by the statement (enter FORTRAN deck or decks here).

Estimates for the CPU time (in seconds) which is required for problems run on the IBM 360/195 are given by Eqs. (A-2) and (A-3) for steady-state and explicit transient calculations, respectively.

$$(\text{CPU})_{ss} = (5.0 \times 10^{-3})(\text{No. of Nodes})(\text{No. of Iterations}) \quad (A-2)$$

$$(\text{CPU})_t = (2.5 \times 10^{-5})(\text{No. of Nodes})(\text{No. of Time Steps}) \quad (A-3)$$

For the implicit transient algorithm, the amount of CPU time per time step will be much larger than that for one of the explicit techniques since the code must do a lot more computation. However, one should be able to use a time step much larger than would be required in the explicit technique, especially for later times in the transient. Frequently, the implicit algorithm is more than an order of magnitude faster than the explicit algorithm in solving a problem. Temperature-dependent thermal properties and/or boundary conditions can increase the running time by up to an order of magnitude depending on how many nodes are associated with the temperature-dependent parameters. Equations (A-2) and (A-3) were derived empirically by examining a large number of cases which were run on the HEATING4 code, a code with similar calculational techniques. It is emphasized that these equations are merely estimates, and the actual CPU time may be considerably different from the estimate depending on the options which are used.

```

C =====
C ===== HEXEREI2 =====
C ===== HTGR VERSION OF THE HEATING5 CODE WITH HEXAGONAL MESH =====
C =====
C ===== THE HEATING CODE WITH TEMPERATURE-DEPENDENT THERMAL PROPERTIES,
C ===== NON-LINEAR AND SURFACE-TO-SURFACE BOUNDARY CONDITIONS.
C ===== AND AN IMPLICIT TECHNIQUE FOR THE SOLUTION OF TRANSIENT PROBLEMS.
C ===== THIS VERSION IS ESPECIALLY MODIFIED FOR THE HTGR TO INCLUDE
C ===== ONE-DIMENSIONAL COOLANT FLOW IN THE AXIAL DIRECTION FOR 3D RZT
C ===== PROBLEMS WITH HEXAGONAL MESH IN A CYLINDRICAL REGION AROUND THE
C ===== CENTER.
C =====
C*****
C*          LET MAXPTS = MAXIMUM NUMBER OF LATTICE POINTS.
C*          IF ONE WISHES TO CHANGE THE MAXIMUM NUMBER OF LATTICE POINTS,
C*          THEN THE FOLLOWING TWO STATEMENTS MUST BE MODIFIED AS
C*          INDICATED.
C*****
C*    DATA N/MAXPTS/
C*    REAL*8 CORE(25MAXPTS)
C*    DATA  N/765/
C*    REAL*8 CORE(19125)
C*****
C*    INTEGER*2 ICORE(1)
C*    LOGICAL*1 LCORE(1)
C*    EQUIVALENCE (CORE(1),ICORE(1),LCORE(1))
C*    CALL HEXERI(CORE,ICORE,LCORE,N)
C*    STOP
C*    END

```

Fig. A-2. Main Program for HEXEREI2 100 Nodes

```

//jobname JOB (cccc), 'programmer-name'
// *CLASS REGION=500K, CPUxx=yyyz, LINES=///
// EXEC FORTHCLG, REGION.GO=500K, PARM.GO='EU=-1'
// FORT.SYSIN DD *
(Enter FORTRAN deck or decks Here)
/*
// LKED.HEXEREI DD DISP=SHR, DSN=A.GEGHD845.HEXEREI2
// LKED.SYSIN DD *
  INCLUDE HEXEREI2
/*
// GO.FT05F001 DD *
(Insert Data Deck Here)
/*
//

```

Fig. A-3. Format of JCL to Run HEXEREI2 at ORGDP when FORTRAN Deck is Modified

APPENDIX B
NOMENCLATURE

Symbol	Description	Typical Units
g	Mass fraction	Unitless
h	Effective heat transfer coefficient	$\text{Btu}/(\text{hr}\cdot\text{ft}^2\cdot^\circ\text{F})$
h_c	Film coefficient (forced convection)	$\text{Btu}/(\text{hr}\cdot\text{ft}^2\cdot^\circ\text{F})$
h_e	Exponent for natural convection	
h_f	Heat flux	$\text{Btu}/(\text{hr}\cdot\text{ft}^2)$
h_o	Convective heat transfer coefficient from node o to the coolant in the axial channels.	$\text{Btu}/(\text{hr}\cdot\text{ft}^2\cdot^\circ\text{F})$
h_n	Coefficient for natural convection	
h_r	Coefficient for radiation	$\text{Btu}/(\text{hr}\cdot\text{ft}^2\cdot^\circ\text{R}^4)$
k	Thermal conductivity	$\text{Btu}/(\text{hr}\cdot\text{ft}\cdot^\circ\text{F})$
q_{Hi}	Increment heat flow rate	Btu/hr
Δq_i	Net heat flow into node i	Btu
r	Length along R-axis	ft
t	Time	hr
Δt	Time increment	hr
z	Length along Z-axis	ft
A	Cross-sectional area normal to heat flow path	ft^2
A_{co}	Surface area for convective heat transfer from node o to coolant in axial channels.	ft^2
B	Extrapolation factor used in steady-state algorithm	Unitless
B_i	Effective thermal conductance for convection to coolant in axial channels from node i.	$\text{Btu}/(\text{hr}\cdot^\circ\text{F})$
C_i	Thermal capacitance of node i	$\text{Btu}/^\circ\text{F}$
C_p, C	Specific heat	$\text{Btu}/(\text{lb}\cdot^\circ\text{F})$
D_i	See Eq. (2.3.3.8)	
G	Mass flux	$\text{lb}_m/(\text{ft}^2\cdot\text{hr})$

NOMENCLATURE (Contd.)

Symbol	Description	Typical Units
H_i	See Eq. (2.3.3.7)	
I	Total number of nodes whose temperatures must be calculated	
K_{ij}	Effective thermal conductance between nodes i and j	Btu/(hr·°F)
L	Length of heat flow path between nodes	ft
L_i	Number of nodes associated with node i whose number is less than i	
M_i	Number of nodes connected to node i	
N_u	Nusselt number	Unitless
P	Pressure	lb/ft ³
P_i	Power associated with node i	Btu/hr
P_o	Constant factor in definition of input parameter - see Eq. (3.4)	
P_r	Prandtl number	Unitless
Q	Power density	Btu/(hr·ft ³)
R	Gas constant	Unitless
R_e	Reynolds number	Unitless
R_i	Residual at node i in implicit transient algorithm - see Eq. (2.3.3.17)	Btu/hr
T_i	Temperature of node i	°F
T_{ci}	Temperature of coolant node i	°F
V	Volume	ft ³
Z	Factor used in Levy's modification for transient algorithm (Eq. (2.3.3.4))	Unitless
α_m	Number of m^{th} node connected to node i	
β	Acceleration parameter for SOR iteration method for steady-state algorithm	Unitless
ϵ	Convergence criterion for SOR iteration method for steady-state algorithm - see Eq. (2.3.2.7)	Unitless
ϵ_1	Convergence criterion for SOR iteration method for implicit transient algorithm (see Eq. (2.3.3.19))	Btu/hr

NOMENCLATURE (Contd.)

Symbol	Description	Typical Units
ϵ_2	Convergence criterion for SOR iteration method for implicit transient algorithm (see Eq. (2.3.3.20))	Unitless
ϵ_3	Convergence criterion for temperature-dependent iteration scheme for implicit transient algorithm (see Eq. (2.3.3.23))	$^{\circ}\text{F}$
θ	Length along θ -axis; also, temperature ratio	Radians
Θ	Parameter in implicit algorithm for transient problems (see Eq. (2.3.3.16))	Unitless
μ	Viscosity	($\text{lb}_m/\text{hr}\cdot\text{ft}$)
ρ	Density	lb/ft^3
ω	Acceleration parameter for SOR iteration method for implicit transient algorithm	Unitless
<i>Subscripts</i>		
b	Boundary index	
i	The node under consideration	
air	Property pertains to air	
He	Property pertains to helium	
mix	Property pertains to helium-air mix	
j	A neighbor node connected to node i	
<i>Superscripts</i>		
n	Time step number	
(n)	Iteration number	
m	Nonlinear iteration number

APPENDIX C

USER-SUPPLIED SUBROUTINES

Subroutines may be supplied by the user to evaluate any of the parameters listed in Table 3.1. Thus, if an input parameter cannot be defined by one of the built-in functions as described in section 3.5, it is quite simple for the user to add his own computational technique for evaluating the parameter. The user-supplied subroutine is referenced by specifying the parameter as an analytical function and by specifying no coefficients for the corresponding analytical function (i.e., leave the second entry on the A1 card blank and omit the related A2 card). Since this analytical function is only a flag to tell the code to call the appropriate user-supplied subroutine, the same analytical function can be specified for more than one parameter. The computational technique is then programmed in the subroutine associated with the parameter of interest (see Table 3.1), and the job is submitted to the computer according to the format presented in Fig. A-3.

HEXEREI2 contains dummy subroutines for each of the parameters listed in Table 3.1. If the user references one of the routines but fails to supply his own, then the code will write out an error message and stop when that subroutine is called. Each user-supplied subroutine has the same argument list. However, all five independent variables are not initialized for each subroutine. Only the independent variables marked by an x in Table 3.1 are initialized when each respective subroutine is called. Subroutine BNDTMP, a typical, dummy, user-supplied subroutine, is shown in Fig. C-1. With the exception of HEATGN, all the other user-supplied subroutines are basically the same as BNDTMP, the differences being the variables which are initialized (see Table 3.1) and the parameter which is being evaluated. Subroutine HEATGN is shown in Fig. C-2.

If the thermal conductivity of a material is anisotropic, then it must be defined in user-supplied CONDTN (see section 3.6.7).


```

      SUBROUTINE BNDTMP(RVALUE,R,TH,Z,TIM,TSN,VALUE,NUMBER,N)
C *****
C
C WHEN IT IS REFERENCED, THIS SUBROUTINE MUST BE SUPPLIED BY THE USER TO
C EVALUATE THE USER-SUPPLIED FUNCTION DEFINING THE BOUNDARY TEMPERATURE.
C THIS PARAMETER MAY BE TIME-DEPENDENT.
C THE VARIABLES IN THE ARGUMENT LIST ARE DEFINED IN THE FOLLOWING TABLE.
C (Y) INDICATES THAT THE VARIABLE HAS BEEN DEFINED WHEN THIS SUBROUTINE
C IS CALLED. (N) INDICATES IT HAS NOT BEEN DEFINED.
C
C      VARIABLE      TYPE      DEFINITION
C      RVALUE      REAL*8      VALUE OF THE USER-SUPPLIED FUNCTION
C                               EVALUATING THE BOUNDARY TEMPERATURE (THIS
C                               VALUE MUST BE COMPUTED BY THIS ROUTINE). (N)
C                               NAME OF VARIABLE IN CALLING SEQUENCE --
C                               TDUM(I).
C      R           REAL*8      X OR R COORDINATE OF NODE N. (N)
C      TH          REAL*8      Y OR THETA COORDINATE OF NODE N.. (N)
C      Z           REAL*8      Z COORDINATE OF NODE N. (N)
C      TIM         REAL*8      TIME AT WHICH PARAMETER IS TO BE
C                               EVALUATED. (Y)
C      TSN         REAL*8      TEMPERATURE AT WHICH PARAMETER IS TO BE
C                               EVALUATED. (N)
C      VALUE       REAL*8      CONSTANT VALUE OF THE PARAMETER WHICH
C                               APPEARS ON ITS RESPECTIVE INPUT CARD (B1) IF
C                               IT IS NON-ZERO.
C                               IF IT IS ZERO, VALUE CONTAINS 1.0. (Y)
C                               NAME OF VARIABLE IN CALLING SEQUENCE --
C                               BYTEMP(NBDTP).
C      NUMBER      INTEGER*4    INDEX OF PARAMETER BEING EVALUATED. (Y)
C                               NAME OF VARIABLE IN CALLING SEQUENCE - NBDTP
C      N           INTEGER*4    NODE NUMBER. (N)
C *****
C      REAL*8 RVALUE,R,TH,Z,TIM,TSN,VALUE
C      COMMON /INOUT/ IN , IO
C
C INSERT ALGORITHM TO COMPUTE THE BOUNDARY TEMPERATURE FOR BOUNDARY
C CONDITION 'NUMBER' HERE.
C
C      WRITE(IO,9000)
C 9000 FORMAT('1*****'/
C 1 ' YOU HAVE CALLED SUBROUTINE BNDTMP WHICH IS A USER-SUPPLIED FUNC
C 2TION.'/
C 3 ' HOWEVER, YOU HAVE NOT SUPPLIED THIS SUBROUTINE.'/
C 4 ' EITHER SUPPLY SUBROUTINE BNDTMP OR CORRECT THE INPUT DATA SO TH
C 5AT THIS SUBROUTINE IS NOT REFERENCED.')
C      STOP
C      END

```

Fig. C-1. Subroutine BNDTMP.

```

      SUBROUTINE HEATGN(RVALUE,R,TH,Z,TIM,TSN,VALUE,NUMBER,N)
C *****
C
C WHEN IT IS REFERENCED, THIS SUBROUTINE MUST BE SUPPLIED BY THE USER TO
C EVALUATE THE USER-SUPPLIED FUNCTION DEFINING THE VOLUMETRIC HEAT
C GENERATION RATE. THIS PARAMETER MAY BE TIME-, TEMPERATURE- AND/OR
C POSITION-DEPENDENT.
C
C NOTE: IF THE HEAT GENERATION RATE IS WRITTEN AS
C       Q = F(R,TH,Z,TSN) * G(TIM)
C       THEN THIS SUBROUTINE MAY BE CALLED FIRST TO CALCULATE THE
C       FACTOR F. THEN, THIS ROUTINE IS CALLED FROM THE SOLUTION
C       ALGORITHM TO UPDATE THE TIME-DEPENDENT FACTOR G OF
C       THE HEAT GENERATION RATE. WHEN THIS FACTOR IS UPDATED, ONLY
C       TIM, NUMBER, AND VALUE ARE DEFINED WHERE VALUE = 1.0.
C       IF NECESSARY, THE TWO CALLS MAY BE IDENTIFIED BY THE VALUE OF
C       N. IN THE FIRST CALL, N CONTAINS THE NUMBER OF THE NODE
C       WHOSE HEAT GENERATION RATE IS BEING EVALUATED. IN THE SECOND
C       CALL, N IS SET TO ZERO.
C
C THE VARIABLES IN THE ARGUMENT LIST ARE DEFINED IN THE FOLLOWING TABLE.
C (Y) INDICATES THAT THE VARIABLE HAS BEEN DEFINED WHEN THIS SUBROUTINE
C IS CALLED. (N) INDICATES IT HAS NOT BEEN DEFINED.
C
C   VARIABLE   TYPE      DEFINITION
C   RVALUE     REAL*8     VALUE OF THE USER-SUPPLIED FUNCTION
C                                     EVALUATING THE VOLUMETRIC HEAT GENERATION
C                                     RATE (THIS VALUE MUST BE COMPUTED BY THIS
C                                     ROUTINE). (N)
C   R          REAL*8     NAME OF VARIABLE IN CALLING SEQUENCE - GEN.
C   TH         REAL*8     X OR R COORDINATE OF NODE N. (Y)
C   Z          REAL*8     Y OR THETA COORDINATE OF NODE N. (Y)
C   TIM        REAL*8     Z COORDINATE OF NODE N. (Y)
C   TSN        REAL*8     TIME AT WHICH PARAMETER IS TO BE
C                                     EVALUATED. (Y)
C   VALUE      REAL*8     TEMPERATURE AT WHICH PARAMETER IS TO BE
C                                     EVALUATED. (Y)
C   NUMBER     INTEGER*4  CONSTANT VALUE OF THE PARAMETER WHICH
C                                     APPEARS ON ITS RESPECTIVE INPUT CARD (G) IF
C                                     IT IS NON-ZERO.
C                                     IF IT IS ZERO, VALUE CONTAINS 1.0. (Y)
C   N          INTEGER*4  NAME OF VARIABLE IN CALLING SEQUENCE --
C                                     GENO(NGN).
C                                     INDEX OF PARAMETER BEING EVALUATED. (Y)
C                                     NAME OF VARIABLE IN CALLING SEQUENCE - NGN.
C                                     NODE NUMBER. (Y)
C *****
C   REAL*8 RVALUE,R,TH,Z,TIM,TSN,VALUE
C   COMMON /INOUT/ IN , IO
C
C INSERT ALGORITHM TO COMPUTE THE VOLUMETRIC HEAT GENERATION RATE HERE.
C
C   WRITE(IO,9000)
C 9000 FORMAT('1*****'/
C 1 ' YOU HAVE CALLED SUBROUTINE HEATGN WHICH IS A USER-SUPPLIED FUNC
C 2 TION.'/
C 3 ' HOWEVER, YOU HAVE NOT SUPPLIED THIS SUBROUTINE.'/
C 4 ' EITHER SUPPLY SUBROUTINE HEATGN OR CORRECT THE INPUT DATA SO TH
C 5 AT THIS SUBROUTINE IS NOT REFERENCED.')
C   STOP
C   END

```

Fig. C-2. Subroutine HEATGN.

INTERNAL DISTRIBUTION

1. S. J. Ball	16-26. G. E. Giles, Jr.
2. B. R. Becker	27. R. W. Henderson
3. A. A. Brooks	28. R. A. Just
4. J. P. Callahan	29-30. P. R. Kasten
5. Central Research Library	31. G. H. Llewellyn
6. K. W. Childs	32. A. L. Lotts
7. J. C. Cleveland	33-34. D. S. Napolitan
8. J. H. Coobs	35-38. ORGDP Library
9. CSD Library, ORGDP	39. ORGDP Records - RC
10. CSD Library, ORNL	40. H. Postma
11. F. L. Culler	41. M. W. Rosenthal
12. R. M. DeVault	42-46. J. P. Sanders
13. Document Reference Section, Y-12	47. I. I. Siman-Tov
14. G. G. Fee	48. D. B. Trauger
15. Uri Gat	49. W. D. Turner

EXTERNAL DISTRIBUTION

50. Director, Division of Nuclear Research and Application, ERDA, Washington, D.C. 20555
51. Assistant Director, Reactor Programs, NRA, ERDA, Washington, D.C. 20555
52. Chief, Gas-Cooled Thermal Reactors Branch, NRA, ERDA, Washington, D.C. 20555
53. Chief, Fast Gas Reactor Branch, NRA, ERDA, Washington, D.C. 20555
54. Director, Reactor Division, ERDA, ORO
55. Director, Research and Technical Support Division, ERDA, ORO
- 56-103. Technical Information Center, ERDA, ORO (21 copies for ACRS)
104. Assistant Director for Reactor Safety, NRC/NRR/DSS, Washington, D.C. 20555
105. Assistant Director for Advanced Reactors, NRC/NRR/DPM, Washington, D.C. 20555
106. Chief, Reactor System Branch, NRC/NRR/DSS, Washington, D.C. 20555
107. Chief, Analysis Branch, NRC/NRR/DSS, Washington, D.C. 20555
108. Chief, Core Performance Branch, NRC/NRR/DSS, Washington, D.C. 20555
109. Chief, Special Reactors Branch, NRC/NRR/DPM, Washington, D.C. 20555
110. Director, Office of Nuclear Regulatory Research, NRC, Washington, D.C. 20555
111. Assistant Director, Research Coordination, NRC/RES, Washington, D.C. 20555
112. Assistant Director, Advanced Reactor Safety Research, NRC/RES, Washington, D.C. 20555
113. Chief, Experimental Gas-Cooled Reactor Safety Research Branch, NRC/RES, Washington, D.C. 20555
- 114-116. Public Document Room, USNRC, 1717 H St., Washington, D.C. 20555
117. R. E. Ireland, NRC/NRR/DSS, Washington, D.C. 20555
118. M. A. McCoy, NRC/NRR/DSS, Washington, D.C. 20555
119. L. E. Phillips, NRC/NRR/DSS, Washington, D.C. 20555
120. W. Kirk, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87544
121. D. Schweitzer, Brookhaven National Laboratory, Upton, Long Island, NY 11973
- 122-127. General Atomic Company, Library, Room T-109, P.O. Box 81608, San Diego, CA 92138