

2 cys.
RECEIVED BY DTIC JUL 9 1970

Gulf General Atomic
Incorporated

AEC RESEARCH AND
DEVELOPMENT REPORT

GA-9264
████████████████████
████████████████████

MASTER

TAC3D

A GENERAL PURPOSE THREE-DIMENSIONAL
HEAT TRANSFER COMPUTER CODE
MATHEMATICAL FORMULATIONS AND
PROGRAMMER'S GUIDE

by

S. S. Clark, J. V. Del Bene, and J. F. Petersen

Prepared under
Contract AT(04-3)-167
Project Agreement No. 17
for the
San Francisco Operations Office
U.S. Atomic Energy Commission

September 1969

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

[REDACTED]

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.

Gulf General Atomic Incorporated

P. O. Box 608, San Diego, California 92112

AEC RESEARCH AND
DEVELOPMENT REPORT

GA-9264

TAC3D
A GENERAL PURPOSE THREE-DIMENSIONAL
HEAT TRANSFER COMPUTER CODE
MATHEMATICAL FORMULATIONS AND
PROGRAMMER'S GUIDE

Work done by:

S. S. Clark
J. V. Del Bene
R. Katz
J. F. Petersen
M. Troost

Report written by:

S. S. Clark
J. V. Del Bene
J. F. Petersen

Prepared under
Contract AT(04-3)-167
Project Agreement No. 17
for the
San Francisco Operations Office
U.S. Atomic Energy Commission

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights, or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

109

The TAC3D computer code is described herein as it existed on September, 1969. The code has been in continuous development for 2 years and in its presented form has been applied successfully by Gulf General Atomic Incorporated to the kind of problems discussed later in this report. However, the development and improvement of the code are being continued, so that duplication of results (or even close agreement) between problems run with the code as published and the code as it existed either before or after this time is not necessarily to be expected.

Gulf General Atomic has exercised due care in preparation but does not warrant the merchantability, accuracy, and completeness of the code or of its description contained herein. The complexity of this kind of program precludes any guarantee to that effect. Therefore, any user must make his own determination of the suitability of the code for any specific use, and of the validity of the information produced by use of the code.

ABSTRACT

TAC3D is a code for calculating steady-state and transient temperatures in three-dimensional problems by the finite difference method. It is written entirely in Fortran V. The configuration of the body to be analyzed is described in the rectangular or cylindrical coordinate system by orthogonal planes of constant coordinate called grid planes. The grid planes specify an array of nodal elements. Nodal points are defined as lying midway between the bounding grid planes of these elements. A finite difference equation is formulated for each nodal point in terms of its capacitance, heat generation and heat flow paths to neighboring nodal points. A system of these equations is solved by an implicit method which is one of the most efficient known at this time.

Some advantages of the code are:

1. The geometrical input is simple.
2. The input of thermal parameters is by Fortran V arithmetic statement functions. Many of the calculation variables (time, local temperature, local position, etc.) are available for use in these functions.
3. Internal and external flowing coolants may be used.
4. There may be internal and external thermal radiation.
5. There is a wide selection of optional output.

The principal limitations of the code are:

1. The grid plane system must be orthogonal in the rectangular or cylindrical coordinate system. Therefore, the sides of the nodal elements must also be orthogonal. The entire

problem must be bounded by six grid planes in one of the coordinate systems. Difficulties in treating irregular boundaries can be overcome to some extent through the use of materials having specially chosen properties.

2. All radiation is treated one-dimensionally.
3. There are no provisions for thermal expansion or change of phase. Such special heat transfer situations could be included by extensions of the existing programming.

TAC3D has been assigned operational status. The machine requirement is a 65K Univac 1108, or equivalent. In addition to input-output, a maximum of four and a minimum of no external storage devices are required depending upon the code options being used. The operating system under which the code has been successfully used is EXEC II as modified for Gulf General Atomic. Running time depends upon the size and complexity of the problem and is not easily defined.

A related code TAC2D, is two-dimensional and has all the features of TAC3D described in this abstract. The Fortran decks, test cases, a user's manual and a descriptive report are available through Argonne Code Center for each of the two codes.

TABLE OF CONTENTS

ABSTRACT	iii
TABLE OF CONTENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	vii
1. INTRODUCTION	1
2. GEOMETRICAL CONSIDERATIONS	5
2.1 GRID PLANES, POINTS, AND BOUNDARIES	5
2.2 BLOCKS	8
2.2.1 General	8
2.2.2 Coolant Blocks	11
2.3 GAPS	11
2.4 GAP PLANES	12
3. MATERIAL, GAS AND COOLANT THERMAL PARAMETERS	13
3.1 DESCRIPTION OF THERMAL PARAMETER FUNCTIONS	13
3.2 EVALUATION OF THERMAL PARAMETERS	14
3.3 FUNCTION CONTROL CONSTANTS	15
4. GENERAL CODE DESCRIPTION	17
4.1 PROGRAM SECTIONS	17
4.1.1 Input Processing Section	17
4.1.2 Computational Section	18
APPENDIX A - MATHEMATICAL SOLUTION	21
APPENDIX B - GEOMETRY FACTORS	27
APPENDIX C - COMPOSITE CONDUCTIVITIES	31

Table of Contents (Continued)

C.1	NO GAP BETWEEN ADJACENT MESH POINTS, EQUAL THERMAL CONDUCTIVITIES	32
C.2	NO GAP BETWEEN ADJACENT MESH POINTS, UNEQUAL THERMAL CONDUCTIVITIES	32
C.3	A GAP BETWEEN ADJACENT MESH POINTS	35
	C.3.1 Iterative Evaluation of Gap Surface Temperatures . .	36
	C.3.2 Geometrical Constants for Gaps	37
C.4	COOLANTS	41
	C.4.1 Coolant on High Index Side of Grid Plane	42
	C.4.2 Coolant on Low Index Side of Grid Plane	44
	C.4.3 Geometrical Constants for Coolants	45
	C.4.4 Two Coolants Adjacent	46
APPENDIX D	- COOLANTS	47
APPENDIX E	- STEADY-STATE OPTION	53
APPENDIX F	- SAMPLE PROBLEM	57
	F.1 PROBLEM DESCRIPTION	57
	F.2 ANALYTICAL SOLUTION	58
	F.3 NUMERICAL SOLUTION WITH TAC3D	58
	F.4 COMPARISON OF THE TAC3D SOLUTION WITH THE ANALYTICAL SOLUTION	60
APPENDIX G	- CODE ORGANIZATION AND FUNCTIONS OF SUBROUTINES IN TAC3D	67
APPENDIX H	- TAC3D CODE LISTING	75
REFERENCES	242

LIST OF TABLES

Table 1.	Block Boundary and Block Point Nomenclature	8
Table 2.	Internal Block Definition	10
Table 3.	External Block Definition	10
Table C.1.	Geometrical Constant $(\Delta X_1/A_1)/(\Delta X/A)$	35
Table C.2.	Geometrical Constant $(A_2/\Delta X_2)/(A/\Delta X)$	40
Table C.3.	Geometrical Constant $(A_2/\Delta X_2)/(A_1/\Delta X_1)$	40
Table C.4.	Geometrical Constant $A_g/(A_2/\Delta X_2)$	40
Table F.1.	Input Thermal Parameter Functions for TAC3D Sample Problem	61
Table F.2.	Excerpts from TAC3D Sample Problem Output Showing Block, Grid Plane and Time History Specifications	62

LIST OF FIGURES

Fig. 1.	Axial Plane of a Typical Rectangular Geometry Envelope Showing Grid Planes and Mesh Points	6
Fig. 2.	Axial Plane of a Typical Cylindrical Geometry Envelope Showing Grid Planes and Mesh Points	7
Fig. 3.	Axial Plane of a Typical Problem Model in Cylindrical Geometry	9
Fig. 4.	General Flow of Logic in TAC3D	19
Fig. C.1.	Logical Search at Each Grid Plane	32
Fig. F.1.	TAC3D Sample Problem — Thermal Model	59
Fig. F.2.	TAC3D Sample Problem — Comparison of Code Solution With Analytical Solution	66

1. INTRODUCTION

The purpose of this report is to document methods of solution and programming logic for TAC3D.* All information required for use of the code is given in Reference 1.

The TAC3D code is designed to treat transient, three-dimensional heat transfer problems. Steady-state problems are treated by considering the problem to be a transient, starting with an assumed temperature distribution and running until equilibrium conditions are established. The code includes a special option which may be used to perform such calculations efficiently. Geometrically, the problem may be defined by either rectangular (X, Y, Z) or cylindrical (R, Z, θ) coordinates. Although there are a number of three-dimensional heat transfer codes, it was decided to write a new code because the existing codes have at least one of the following limitations:

1. Long computational time
2. Complicated input (data for each point, explicitly)
3. Constant properties (conductivities not temperature dependent)
4. No radiation across internal gaps
5. No internal coolants

The form of the code was dictated by the method of solution chosen for the system of simultaneous finite difference equations. This is an implicit alternating direction scheme which requires a "regular" geometry in that the points at which temperatures are to be

*The acronym TAC3D stands for "Thermal Analysis Code - Three Dimensional."

calculated must be in regular rows, columns, and planes. As a consequence, TAC3D is primarily suited to solve problems that roughly fit an envelope of either a rectangular parallelepiped or an incomplete right circular cylinder.

The basic heat transport equation to be solved is

$$\nabla \cdot k\nabla T + S = \frac{\partial}{\partial \tau} \rho c T$$

where

k is thermal conductivity, Btu/hr-ft-°F

T is local temperature, °F

S is volumetric heat generation rate, Btu/hr-ft³

ρ is density, lb/ft³

c is specific heat, Btu/lb-°F

τ is time, hr

This equation is replaced by its equivalent set of linear finite difference equations. This set of equations is then solved by an implicit alternating direction method which is described in Appendix A. The derivation of the proper form of $\nabla \cdot k\nabla T$ is described in Appendices B and C.

The problem, whose temperature distribution as a function of time we wish to obtain, becomes a set of mesh points and boundaries to TAC3D. Heat transfer across internal boundaries may be by radiation, conduction, and convection. Heat transfer across external boundaries is by convection only. This is not a restriction since insulated boundaries, constant temperature boundaries, convection, and radiation can all be accommodated by the proper specification of the thermal parameters (e.g., heat transfer coefficient, flow rate, inlet temperature and specific heat) of the coolants which cool the external boundaries. Coolants are discussed in Appendix D.

The TAC3D code has great versatility because of its ability to accept material gas and coolant thermal parameters in a functional form.

Axial conductivity, radial conductivity, theta conductivity, heat generation, thermal emissivities in the three coordinate directions, and specific heat may be specified for materials. These thermal parameters may be functions of all of or some of the variables temperature, time, and location. For a gas present in an internal gap, the gas conductivity, which may be both temperature and location dependent, is specified.

Coolants may be present both internally and externally. For coolants, the specific heat, Reynolds number, heat transfer coefficient, flow rate, and inlet temperature may be specified. These thermal parameters may be functions of all of or some of the following variables: location, time, bulk temperature, local surface temperature, Reynolds number, flow rate, inlet temperature, and outlet temperature.

Because of the ability to subdivide a problem into blocks, each containing a material or a coolant with its specified thermal parameters, even complicated geometries can be handled conveniently.

The number of different materials, gases and coolants are limited to fifteen of each. There are also limitations on the number of mesh points, internal gaps and coolant boundaries. The maximum numbers are discussed in Appendix E of Reference 1.

Another limitation is that radiation is assumed to travel straight across each internal gap and coolant passage. In other words, the geometry factor is assumed to be unity. For narrow gaps and coolant passages this assumption is good, but for wide gaps and wide internal coolant channels it may not be correct.

TAC3D is actually one of two generalized heat transfer codes which have been developed at Gulf General Atomic. The other code is TAC2D which is a two-dimensional version of TAC3D and is described in Refs. 2 and 3.



✓

6

✓

•



2. GEOMETRICAL CONSIDERATIONS

2.1 GRID PLANES, POINTS, AND BOUNDARIES

Two geometry types are available to describe the model--cylindrical and rectangular. The model envelope and coordinate axes are determined from this choice.

<u>Model Envelope</u>	<u>Coordinate Axes</u>
Rectangular parallelepiped	X, Y, Z
Incomplete* right circular cylinder ($0 \leq \theta < 2\pi$)	R, Z, θ

From this point on, the words radial, axial, and theta will be used for the coordinate axes. When setting up a problem model in rectangular geometry, the user may equate X, Y, Z with radial, axial, and theta, as he chooses.

In each of the coordinate directions, a set of grid planes, not necessarily evenly spaced, is specified to define a three-dimensional "grid" mesh which bounds and divides the envelope. From this set of grid planes, an auxiliary mesh, the "points" mesh, is created by locating points centered between grid planes and on outside boundary grid planes. It is at these points that the temperature distribution is calculated. Typical cross sections of the meshes in axial planes of the envelopes are shown for the rectangular and cylindrical geometry types in Figs. 1 and 2, respectively. A distinction between "internal" points and "external" points is shown in the figures. The external points represent the external coolants.

In Figs. 1 and 2, TL(k) and RL(i) label the grid planes in the theta and radial directions, respectively, and TP(k) and RP(i)

*The cylinder is termed incomplete because, although the coordinate θ may extend from 0° through 360° , there is no connection between mesh points adjacent to the grid planes $\theta = 0^\circ$ and $\theta = 360^\circ$.

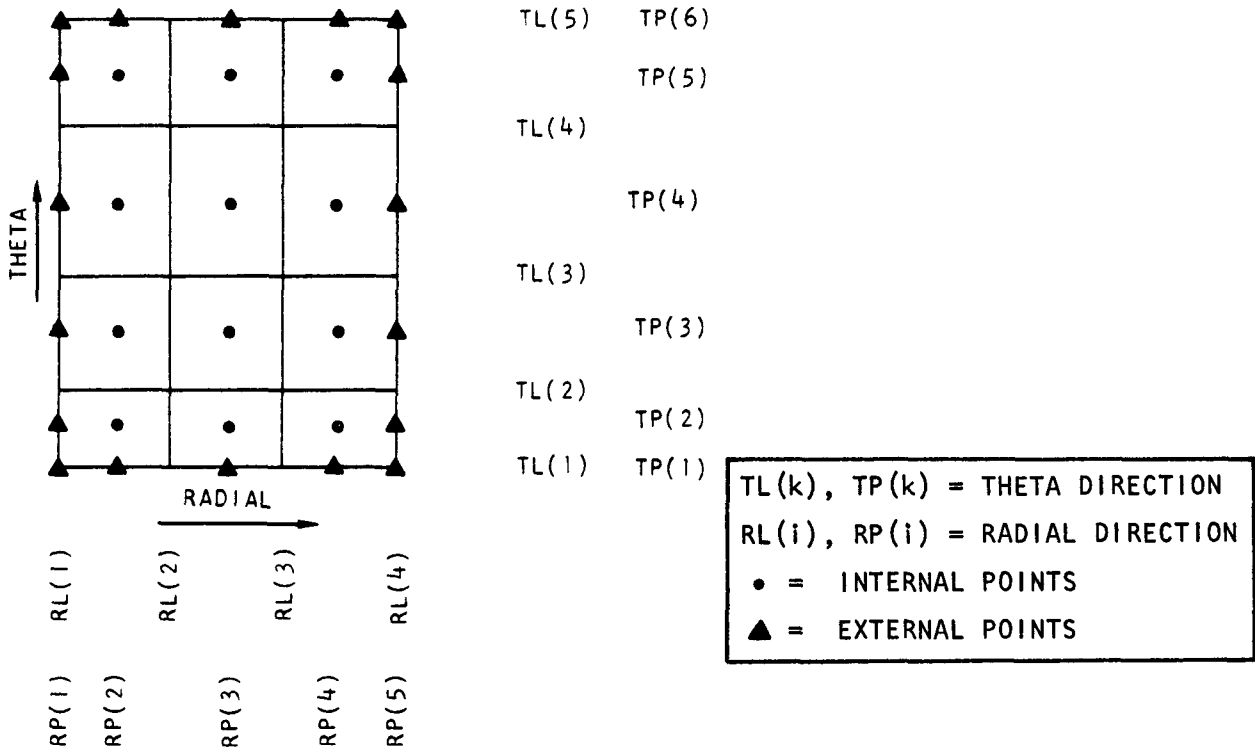
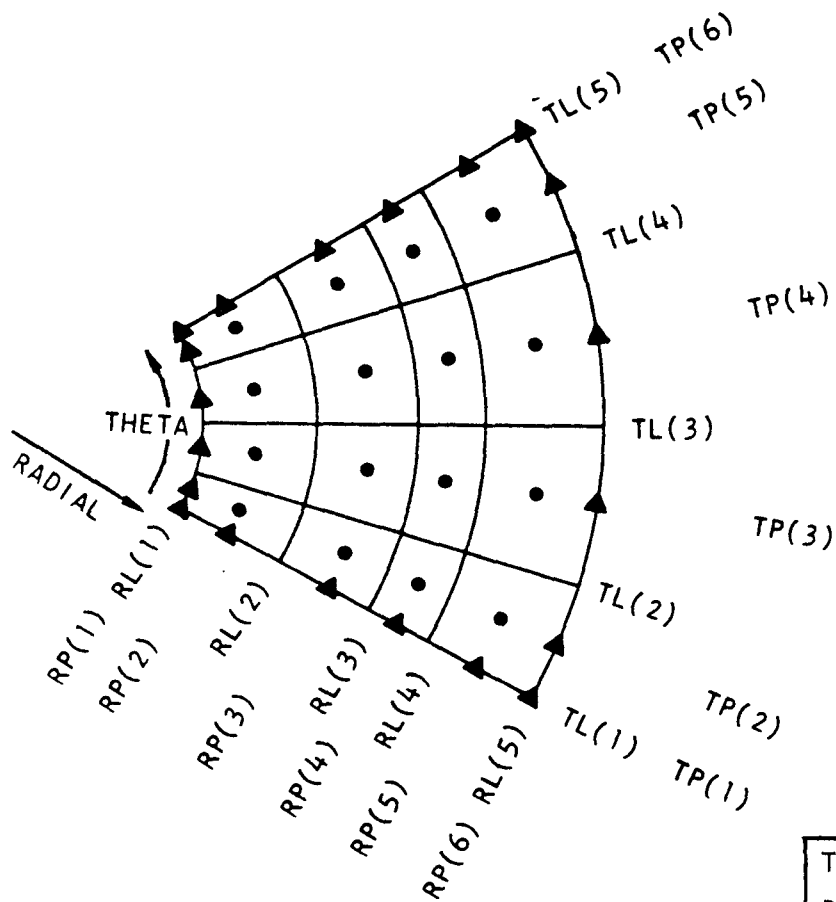


Fig. 1. Axial plane of a typical rectangular geometry envelope showing grid planes and mesh points



<p> $TL(k), TP(k)$ = THETA DIRECTION $RL(i), RP(i)$ = RADIAL DIRECTION ● = INTERNAL POINTS ▲ = EXTERNAL POINTS </p>
--

Fig. 2. Axial plane of a typical cylindrical geometry envelope showing grid planes and mesh points

label the mesh points in the theta and radial directions, respectively.

2.2 BLOCKS

2.2.1 General

The envelope may be subdivided into a number of blocks. A block is a rectangular parallelepiped or an incomplete right circular cylinder. Each of the material or coolant thermal parameters must be defined by the same function at all points within a block; i.e., a block consists of only one homogeneous material or coolant. Each block has six bounding grid planes which determine the mesh points associated with the block. The boundary and point nomenclature of the block is given in Table 1.

Table 1
BLOCK BOUNDARY AND BLOCK POINT NOMENCLATURE

	Coordinate Direction	Low Boundary	High Boundary
Block boundaries	Radial	ILS	IHS
	Axial	JLS	JHS
	Theta	KLS	KHS
Block points	Radial	ILS+1	IHS
	Axial	JLS+1	JHS
	Theta	KLS+1	KHS

Figure 3 gives an example of a problem model with its enclosing envelope. The TAC3D envelope chosen for the model is not completely filled by the model because a cylindrical geometry was selected. The part of the envelope which does not contain the model contains a "dummy material." The model boundary has to be approximated by using orthogonal grid planes. In Fig. 3, TL(k) and RL(i) label the grid planes in the theta and radial directions, respectively. A set of blocks which could be used to subdivide the envelope and specify its external boundary conditions is given in Tables 2 and 3.

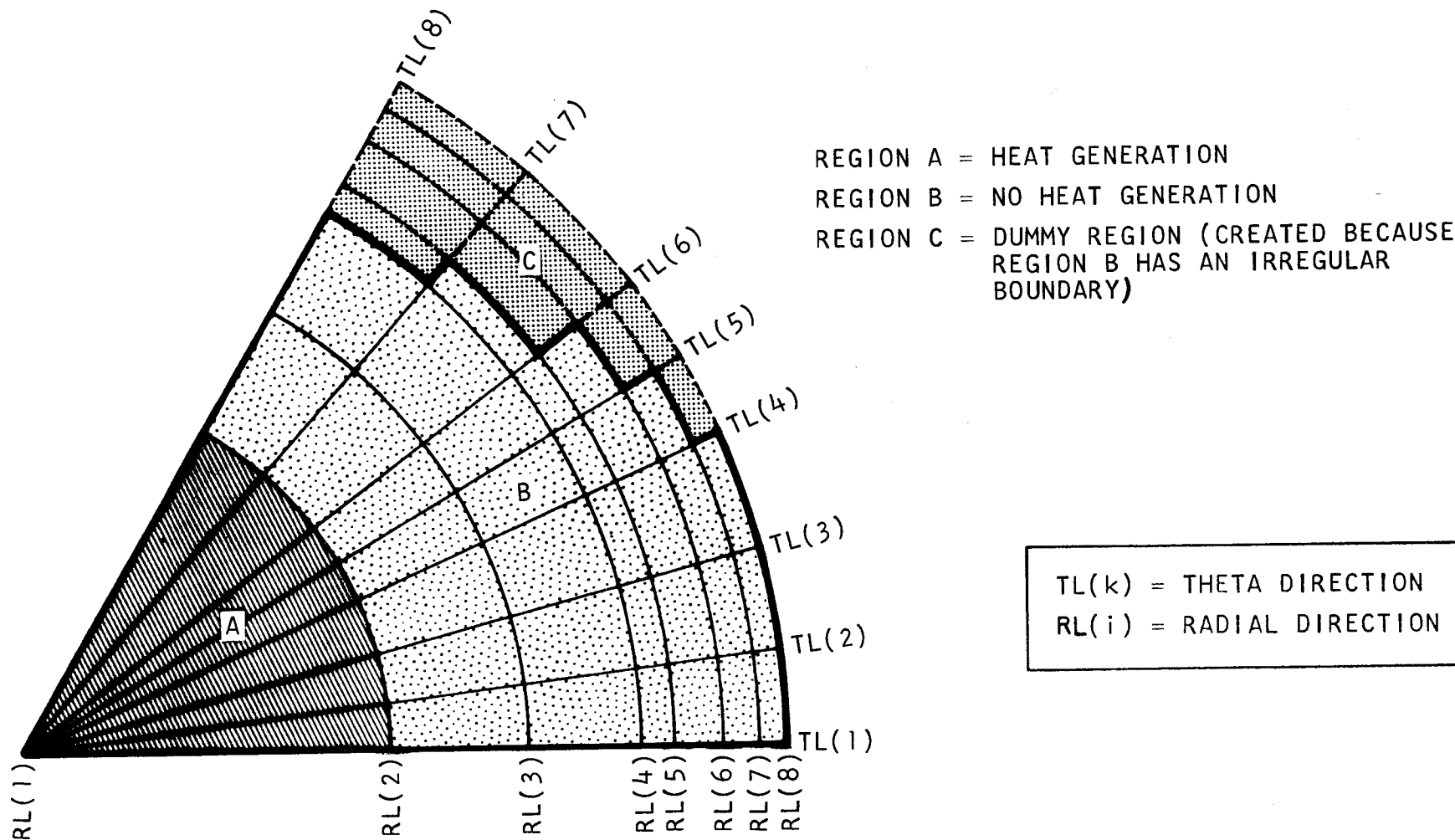


Fig. 3. Axial plane of a typical problem model in cylindrical geometry

TABLE 2
INTERNAL BLOCK DEFINITION

Block Number	Block Grid Plane Boundaries		
	Radial	Theta	Axial
1 (Region A)	RL(1)-RL(2)	TL(1)-TL(8)	ZL(1)-ZL(J)
2 (Region B)	RL(2)-RL(8)	TL(1)-TL(4)	ZL(1)-ZL(J)
3 (Region B)	RL(2)-RL(7)	TL(4)-TL(5)	ZL(1)-ZL(J)
4 (Region B)	RL(2)-RL(6)	TL(5)-TL(6)	ZL(1)-ZL(J)
5 (Region B)	RL(2)-RL(5)	TL(6)-TL(7)	ZL(1)-ZL(J)
6 (Region B)	RL(2)-RL(4)	TL(7)-TL(8)	ZL(1)-ZL(J)
7 (Dummy)	RL(7)-RL(8)	TL(4)-TL(8)	ZL(1)-ZL(J)
8 (Dummy)	RL(6)-RL(7)	TL(5)-TL(8)	ZL(1)-ZL(J)
9 (Dummy)	RL(5)-TL(6)	TL(6)-TL(8)	ZL(1)-ZL(J)
10 (Dummy)	RL(4)-RL(5)	TL(7)-TL(8)	ZL(1)-ZL(J)

TABLE 3
EXTERNAL BLOCK DEFINITION

Block Number (Boundary)	Block Grid Plane Boundaries		
	Radial	Theta	Axial
11 (low radial)	RL(1)-RL(1)	TL(1)-TL(8)	ZL(1)-ZL(J)
12 (part high radial)	RL(8)-TL(8)	TL(1)-TL(4)	ZL(1)-ZL(J)
13 (part high radial)	RL(8)-RL(8)	TL(4)-TL(8)	ZL(1)-ZL(J)
14 (low theta)	RL(1)-RL(8)	TL(1)-TL(1)	ZL(1)-ZL(J)
15 (high theta)	RL(1)-RL(8)	TL(8)-TL(8)	ZL(1)-ZL(J)
16 (low axial)	RL(1)-RL(8)	TL(1)-TL(8)	ZL(1)-ZL(1)
17 (high axial)	RL(1)-RL(8)	TL(1)-TL(8)	ZL(J)-ZL(J)

The internal blocks are given in Table 2. It is assumed that these blocks extend over the entire axial length of the envelope and that the envelope has been subdivided by a number of axial grid planes, J, which include its external boundaries.

The external surfaces of the envelope also need to be covered by blocks to set up the boundary conditions. These external blocks are

given in Table 3 and are defined by specifying two identical grid planes in one coordinate direction.

2.2.2 Coolant Blocks

Coolant block temperatures are determined by a finite difference solution of the heat balance equation

$$dq = WC_p dT_c$$

where

q = heat transferred to the coolant from adjacent material points, Btu/hr

W = coolant mass flow rate, lb/hr

C_p = coolant specific heat capacity (constant pressure), Btu/lb-°F

T_c = coolant temperature, °F

Coolants must flow parallel to one of the three coordinate axes. The flow direction may be either positive (in the direction of increasing coordinate value) or negative.

Perfect mixing is assumed for all coolants. Therefore, all points lying on the same plane perpendicular to the flow direction in an internal coolant block will be at the same temperature.

Heat transfer in coolant blocks occurs only at the block boundaries which are parallel to the flow direction and only in a direction perpendicular to the boundary at which heat is being transferred. For external blocks the heat transfer is by convection at the external boundaries of the problem. For internal coolant blocks it is by convection at the coolant block boundaries and by radiation between opposite coolant block boundaries.

2.3 GAPS

Within the model, which has now been reduced to a set of blocks, small spaces may be present. These spaces are assumed to be small enough that it would be impractical to bound them by grid planes, thereby making them blocks. The spaces are called "gaps" and are created by removing a layer of material from the block adjacent to the

high index boundary of the block. A block may have a gap on any one or all of its high (outer) radial, axial and theta boundaries, provided the boundary is not an external boundary of the problem and provided it is not shared with an adjacent block which contains a coolant.

Heat is transferred across a gap by conduction through the gas and by radiation between its bounding surfaces. This heat transfer is one-dimensional and occurs only in a direction normal to the block boundary upon which the gap is specified.

The code includes secondary corrections which are applied where gaps are present. They are:

1. Gap volumes are subtracted from node volumes in determining nodal heat capacities and heat generation rates. See Appendix B.
2. When a gap lies parallel to the heat flow direction, its area is subtracted from the heat flow area between two adjacent mesh points. In some cases, this correction is only approximate. See Appendix B.
3. When a gap lies perpendicular to the heat flow direction, its thickness is subtracted from the conduction path length between the block boundary where the gap is specified and the adjacent mesh point. See Appendix C, Eq. (C-10).

2.4 GAP PLANES

A gap plane is defined as a special case of a grid plane. A grid plane is also a gap plane when any region of it is adjacent to a coolant block or to a gap. It follows that the external boundaries of the problem are always gap planes because they are adjacent to the external coolants which must always be present.

3. MATERIAL, GAS AND COOLANT THERMAL PARAMETERS

3.1 DESCRIPTION OF THERMAL PARAMETER FUNCTIONS

The material or coolant in each block is identified by a number. These numbers are 1 through 15 for materials and -1 through -15 for coolants. Each number denotes a discrete set of thermal parameters. The gas in each gap is identified by a number 1 through 15 although for gases there is only one thermal parameter — the thermal conductivity.

The thermal parameters are available as functional relationships. In TAC3D, one must supply them as FORTRAN V arithmetic statement functions. They are compiled into the program as will be described in Section 3.2. Any FORTRAN V arithmetic statement function is allowable, including the use of previously defined arithmetic statement functions or function sub-programs. Each statement function may use any variable in COMMON. Because certain subscripted variables are frequently needed, a number of non-subscripted variable names in COMMON have been set aside to contain their current local values. For instance, the current local value of a mesh point is assigned to the variable named DR, which may be used in certain arithmetic statement functions. The values of all thermal parameters are re-evaluated for every time step. When the user does not specify a certain thermal parameter, a standard value which is present in the code is used. The names and definitions of the functions, their allowed variables and also their standard values are given in Section 2.5 of the user's manual (Ref. 1). They are not repeated here since they are of interest primarily to the code user.

3.2 EVALUATION OF THERMAL PARAMETERS

3.2.1 Overall Logic

Throughout problem execution, Subroutine BLOCK* is called during the computation over each time step. BLOCK calls Subroutine MADATA for each material block and Subroutine FLODAT for each coolant block. MADATA determines current local values for all thermal parameters of the material in a material block. FLODAT makes the same determination for the coolant in a coolant block. The number of evaluations of each thermal parameter within a block depends upon the nature of the parameter and its allowed variables. For instance, the material thermal conductivity in either coordinate direction is a physical property which is relevant to mesh points and may be dependent upon both point temperature and location. Therefore, it is evaluated for all points within the block. The inlet temperature of a coolant, on the other hand, is relevant only to the coolant block as a whole. Therefore, although it may be dependent upon variables such as flow rate and outlet temperature, only one value is determined for the entire block. BLOCK is called only once during the computation for a given time step. All material and coolant parameter values are obtained through this call and stored.

For a gas in a gap there is only one thermal parameter — the thermal conductivity. MADATA is called by Subroutine CONDUC to obtain the current local value. This is done during the calculation of composite conductivities between points adjacent to gaps. The gas conductivity for each point level along each gap is obtained by an individual call to MADATA at the time the value is actually required in the calculation.

3.2.2 Function Usage

The values assigned to the thermal parameters are obtained in MADATA and FLODAT by reference to functions. This process is explained through the following example. Consider a material block containing a material number 1. Let us say that k_r , its thermal conductivity in the radial

*See Appendices G and H for descriptions and listings, respectively, of the subroutines whose names are used in the following text.

direction, is the following function of T, which is its local temperature in °F.

$$k_r = 27.0 - (5.16 \times 10^{-3}) (T)$$

Then, according to the nomenclature of Ref. 1, Section 2.5, the desired input is the arithmetic statement function:

$$RCON1(X) = 27.0 - 5.16E-3 * (DR - 460.0) \quad .$$

Subroutine MADATA is recompiled for this particular problem with the above function being inserted somewhere before its first executable statement. Upon problem execution, MADATA is called as described in Section 3.2.1 for a material block composed of Material 1. Within MADATA, RCON1(X) is recognized as an internal function and becomes the reference used to calculate all current local values of radial thermal conductivity within that block. If no function RCON1(X) is compiled into MADATA, then the value of RCON1(X) is sought from an external reference. The value found is 0.0 which is contained in Function Subprogram FMAT1*. There are fifteen such function subprograms, FMAT1 through FMAT15, with one corresponding to each of Materials and Gases 1 through 15. They have multiple entry points which are the material and gas thermal parameter function names.

Evaluation of gas thermal conductivities in MADATA and of coolant thermal parameters in FLODAT is carried out in the manner illustrated above for materials. The external references for FLODAT are the multiple entry points of Function Subprograms COOL1 through COOL15. One of these function subprograms corresponds to each of Coolants 1 through 15.

3.3 FUNCTION CONTROL CONSTANTS

The need to recompile MADATA and FLODAT for each set of thermal parameter function input imposes a limitation on the use of the code to run consecutive problems. This limitation can be partially overcome through the use of the constants A1 through A18, which are read in as data and are included in COMMON. To illustrate, the example function of Section 3.2.2 could have been written

$$RCON1(X) = A1 - A2 * (DR - 460.0)$$

*In MADATA, any thermal conductivity of 0.0 is reset to 10^{-6} Btu/hr-°R so that the latter is the value actually used in calculations.

with A1 and A2 defined as 27.0 and 5.16×10^{-3} , respectively, in the first set of problem input. Both A1 and A2 could then have been redefined in the input of a consecutively run problem to yield, effectively, a different function for RCON1(X).

4. GENERAL CODE DESCRIPTION

4.1 PROGRAM SECTIONS

TAC3D is composed of two principal sections. They are:

1. Input Processing Section
2. Computational Section

A brief description of each section follows.

4.1.1 Input Processing Section

The finite difference description of a heat transfer problem requires that the problem envelope be divided into small units of material. It is assumed that within each of these units, the thermal parameters and temperature are essentially uniform. To establish these units, the problem envelope is divided by grid planes creating unit cells. The envelope is then more coarsely divided into blocks which are groups of unit cells, all with the same material and material thermal parameter dependence.

Within the input processing section, data setting up these envelope divisions are read and analyzed. Gaps and coolants are assigned. Required initializations are performed. All of the geometric data are checked, and, if they are found incorrect or incompatible, appropriate error messages are printed and the computation is eventually terminated. Certain geometrical constants are calculated, and an initial temperature distribution and problem time history are read.

4.1.2 Computational Section

Sets of temperature calculations called iterations are performed. These are based upon an input of initial mesh point temperatures and one of the following:

1. Prescribed time steps given as input.
2. Code generated time steps.

The latter are used in conjunction with the steady-state option described in Appendix E. A skeleton outline of the flow of logic in the computational section is shown in Fig. 4. The code logic is described in greater detail in Appendix G.

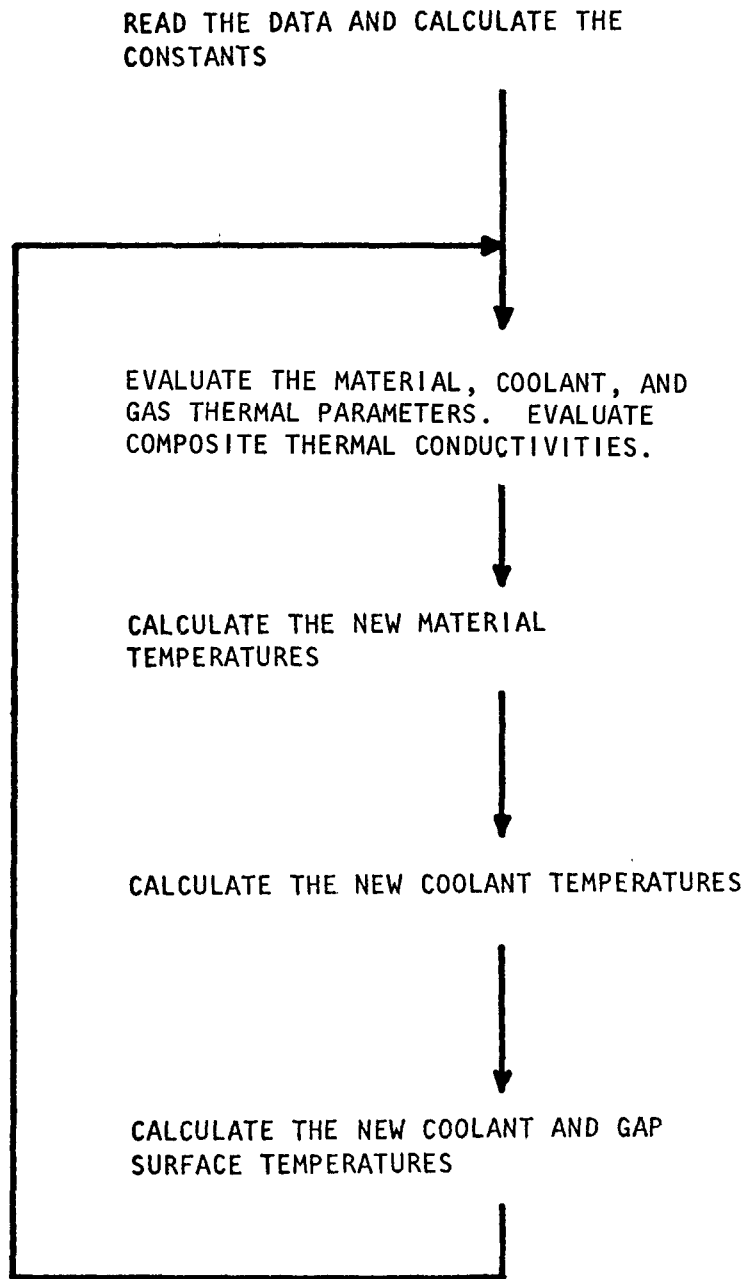


Fig. 4. General flow of logic in TAC3D



APPENDIX A
MATHEMATICAL SOLUTION

The time dependent heat conduction differential equation is

$$\nabla \cdot k_{xyz} \nabla T + S_{xyz} = (\rho c)_{xyz} \frac{\partial T}{\partial \tau} , \quad (A-1)$$

where

k = thermal conductivity in each of the 3 dimensions

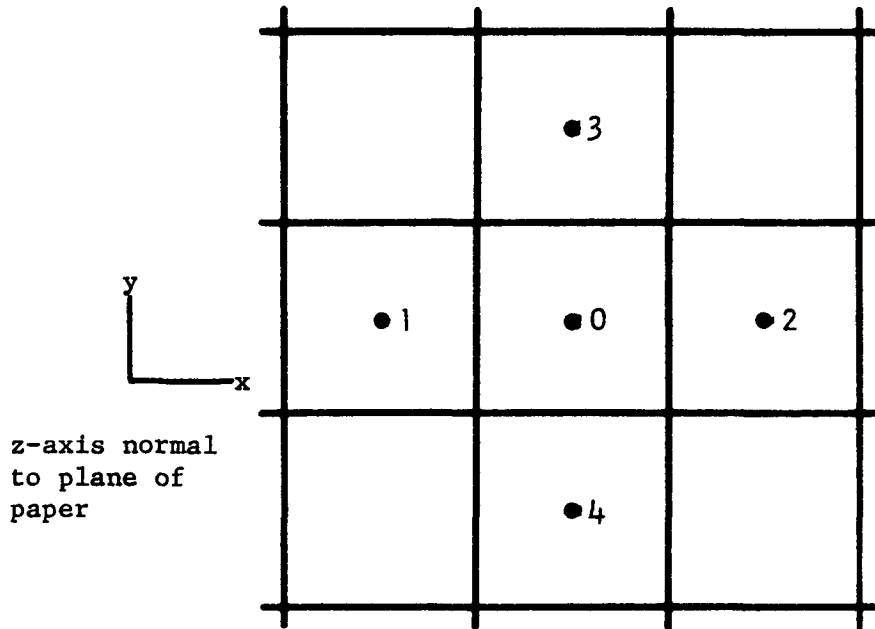
T = temperature ($^{\circ}$ R)

S = heat generation (Btu/ft³-hr)

ρc = volumetric specific heat (Btu/ft³- $^{\circ}$ R)

τ = time (hr)

In the following discussion, Eq. (A-1) is solved by finite difference methods. Consider the envelope of the problem volume as a set of unit cells, each containing a "mesh point" in its center. For instance, the unit cell surrounding point 0 is sketched on the next page. Points five and six would be located above and below the plane of the paper.



The finite difference equation expressing the temperature change in cell 0 in a time period $\Delta\tau$ due to heat moving into or out of the cell or heat being generated within the cell is defined by

$$\begin{aligned}
 & \frac{K_1 A_1}{\Delta X_1} (T_1 - T_0) + \frac{K_2 A_2}{\Delta X_2} (T_2 - T_0) + \frac{K_3 A_3}{\Delta X_3} (T_3 - T_0) \\
 & + \frac{K_4 A_4}{\Delta X_4} (T_4 - T_0) + \frac{K_5 A_5}{\Delta X_5} (T_5 - T_0) + \frac{K_6 A_6}{\Delta X_6} (T_6 - T_0) \\
 & + S_0 V_0 = \frac{(T'_0 - T_0) \rho c V_0}{\Delta\tau}
 \end{aligned} \tag{A-2}$$

where

A_i = average cell surface area between point 0 and i ($i = 1, 6$)

ΔX_i = distance between points 0 and i

K_i = composite thermal conductivity between points 0 and i
in the principal coordinate directions

T_i = temperature of point i ($^{\circ}\text{R}$)

T'_0 = temperature of point 0 ($^{\circ}\text{R}$) after a period of time $\Delta\tau$

S_0 = heat generation per unit volume in cell 0

V_0 = corrected cell volume

The definition of the geometrical quantities $A_i/\Delta X_i$ and V_0 is discussed in Appendix B, and the derivation of the composite conductivity K_i from the local thermal conductivities k_i and k_{i+1} is discussed in Appendix C. In the manner used for point 0, we create as many finite difference equations as there are mesh points. Of the many possible methods for solving such a system of simultaneous equations, an implicit alternating direction method suggested by Douglas in Ref. 4 was selected. This method is only valid for linear equations but may be used here by transforming the nonlinear system (material properties may be temperature dependent) into a quasi-linear system in which the nonlinear factors are frequently re-evaluated.

A short derivation of the method of Douglas is presented below. In this method, the total time step $\Delta\tau$ is divided into three equal subintervals $\delta\tau$. For each subinterval, Eq. (A-2) is written implicitly in one direction and explicitly in the remaining two directions. Defining T' , T'' , and T''' as the temperatures at the times $\tau+\delta\tau$, $\tau+2\delta\tau$, and $\tau+3\delta\tau$, respectively. Eq. (A-2) is rewritten in the following three forms:

$$\begin{aligned} & C_1 \left(\frac{(T'_1 + T_1)}{2} - T_0 \right) + C_2 \left(\frac{(T'_2 + T_2)}{2} - T_0 \right) + C_3 (T_3 - T_0) + C_4 (T_4 - T_0) \\ & + C_5 (T_5 - T_0) + C_6 (T_6 - T_0) = C_7 (T'_0 - T_0) - W \end{aligned} \quad (\text{A-3})$$

and

$$\begin{aligned}
& c_1 \left(\frac{(T_1' + T_1)}{2} - T_0 \right) + c_2 \left(\frac{(T_2' + T_2)}{2} - T_0 \right) + c_3 \frac{(T_3'' + T_3)}{2} - T_0 \quad (A-4) \\
& + c_4 \left(\frac{(T_4'' + T_4)}{2} - T_0 \right) + c_5 (T_5 - T_0) + c_6 (T_6 - T_0) = c_7 (T_0'' - T_0) - W
\end{aligned}$$

and

$$\begin{aligned}
& c_1 \left(\frac{(T_1' + T_1)}{2} - T_0 \right) + c_2 \left(\frac{(T_2' + T_2)}{2} - T_0 \right) + c_3 \left(\frac{(T_3'' + T_3)}{2} - T_0 \right) \\
& + c_4 \left(\frac{(T_4'' + T_4)}{2} - T_0 \right) + c_5 \left(\frac{(T_5''' + T_5)}{2} - T_0 \right) + c_6 \left(\frac{(T_6''' + T_6)}{2} - T_0 \right) \\
& = c_7 (T_0''' + T_0) - W \quad (A-5)
\end{aligned}$$

where $C_i = K_i A_i / \Delta x_i$, $W = S_o V_o$ and $C_7 = \rho c V_o / \delta \tau$.

Note, in Eq. (A-3) the unknown temperatures for the time $\tau + \delta \tau$ are along rows parallel to the x-axis; namely, the equation is implicit in this direction and explicit in the other two directions. A similar condition exists for Eqs. (A-4) and (A-5) with regard to the y and z coordinate directions.

Since it is desired to solve the equations line by line using a tri-diagonal solution, the following manipulations are performed. The difference between Eqs. (A-4) and (A-3) and between Eqs. (A-5) and (A-4) are taken to yield, respectively;

$$c_3 \frac{(T_3'' - T_3)}{2} + c_4 \frac{(T_4'' - T_4)}{2} = c_7 (T_0'' - T_0) \quad (A-6)$$

$$c_5 \frac{(T_5''' - T_5)}{2} + c_6 \frac{(T_6''' - T_6)}{2} = c_7 (T_0''' - T_0''). \quad (A-7)$$

Eqs. (A-3), (A-6) and (A-7) are now written in the following form:

$$\frac{C_1 T_1'}{2} - C_7 T_0' + \frac{C_2 T_2'}{2} = -C_7 T_0' - C_1 \left(\frac{T_1}{2} - T_0 \right) - C_2 \left(\frac{T_2}{2} - T_0 \right) \\ - C_3 (T_3 - T_0) - C_4 (T_4 - T_0) - C_5 (T_5 - T_0) - C_6 (T_6 - T_0) - W$$

or

$$AT_1' + BT_0' + CT_2' = D, \quad (A-8)$$

$$\frac{C_3 T_3''}{2} - C_7 T_0'' + \frac{C_4 T_4''}{2} = -C_7 T_0'' + \frac{C_3 T_3}{2} + \frac{C_4 T_4}{2}$$

or

$$AT_3'' + BT_0'' + CT_4'' = D \quad (A-9)$$

and

$$\frac{C_5 T_5'''}{2} - C_7 T_0''' + \frac{C_6 T_6'''}{2} = -C_7 T_0''' + \frac{C_5 T_5}{2} + \frac{C_6 T_6}{2}$$

or

$$AT_5''' + BT_0''' + CT_6''' = D \quad (A-10)$$

Eqs. (A-8), (A-9) and (A-10) form the basis for the alternating direction implicit method. Eq. (A-8) represents an implicit equation for all the temperatures in a single row parallel to the x-axis. By sweeping each of these rows, a new temperature is evaluated at each node for time $\tau + \delta\tau$. Eq. (A-9) is used in sweeping each row parallel to the y-axis to calculate the nodal temperatures at time $\tau + 2\delta\tau$. The last equation is used to calculate the nodal temperatures at the end of the complete time step $\tau + \Delta\tau$.

The alternate sweeping of the tridiagonal set of N equations of the form

$$A_i T_{i-1} + B_i T_i + C_i T_{i+1} = D_i; i=1, N \quad (A-11)$$

is one of the most efficient ways to solve the complete three dimensional system of equations. In solving the set of N equations indicated by Eq. (A-11), the boundary conditions are $A_1 = 0$ and $C_N = 0$. By introducing the temporary quantities E, X_i , Y_i , we may write:
 $E = B_i - A_i X_{i-1}$, $X_i = C_i/E$, and $Y_i = (D_i - A_i Y_{i-1})/E$. After calculating X_i and Y_i from $i = 1$ to $i = N$, we may now calculate the temperatures from $i = N$ to $i = 1$ by the recursion equation $T_i = Y_i - X_i T_{i+1}$.

APPENDIX B

GEOMETRY FACTORS

The overall thermal conductance between mesh points may be factored into two components, one of which depends purely upon geometry and has the general form $A/\Delta X$. The specific forms of this "geometry factor" applicable to TAC3D are derived here in the nomenclature of the code.

In TAC3D, point 0 is a point (I,J,K) whose unit cell is bounded by the following grid planes:

Radial I-1, I
Axial J-1, J
Theta K-1, K

For each unit cell, three coordinate-oriented geometry factors and a volume are calculated. In the Radial direction,

$$\frac{A_1}{\Delta X_1} = RR(I-1, J, K),$$

and

$$\frac{A_2}{\Delta X_2} = RR(I, J, K) \quad (B-1)$$

In the Axial direction,

$$\frac{A_3}{\Delta X_3} = RZ(I, J-1, K),$$

and

$$\frac{A_4}{\Delta X_4} = RZ(I, J, K). \quad (B-2)$$

In the theta direction,

$$\frac{A_5}{\Delta X_5} = RT(I, J, K-1),$$

$$\frac{A_6}{\Delta X_6} = RT(I, J, K), \quad (B-3)$$

$$V_0 = V(I, J, K). \quad (B-4)$$

The nomenclature to be used in the three coordinate directions is as follows:

	<u>Grid Plane</u>	<u>Point</u>	<u>Gap Thickness</u>
Radial	RL	RP	Δg_R
Axial	ZL	ZP	Δg_Z
Theta	TL	TP	Δg_T

Corrections to account for the presence of gaps are included. For the area components, this correction is applied as the arithmetic average of the two Δg values between points in any one coordinate direction. When these values are not the same and do not apply over equal distances, then the correction is only approximate.

The equations given below define the geometry factors for both cylindrical and rectangular geometries. For the cylindrical geometry,

$$RR(I, J, K) = \frac{1}{\left[\ln \left(\frac{RP(I+1)}{RP(I)} \right) \right]} \left[ZL(J) - ZL(J-1) - \frac{\Delta g_Z(I, K) + \Delta g_Z(I+1, K)}{2.0} \right] \\ \times \left[TL(K) - TL(K-1) - \frac{\Delta g_T(I, J) + \Delta g_T(I+1, J)}{2.0} \right] \quad (B-5)$$

$$RZ(I,J,K) = \left\{ \left[RL(I) - \frac{\Delta g_R(J,K) + \Delta g_R(J+1,K)}{2.0} \right]^2 - [RL(I-1)]^2 \right\} \\ \times \left[\frac{TL(K) - TL(K-1) - \left\{ \frac{1}{2.0} [\Delta g_T(I,J) + \Delta g_T(I,J+1)] \right\}}{ZP(J+1) - ZP(J)} \right] \quad (B-6)$$

$$RT(I,J,K) = \left[RL(I) - RL(I-1) - \frac{\Delta g_R(J,K) + \Delta g_R(J,K+1)}{2.0} \right] \\ \times \left[\frac{ZL(J) - ZL(J-1) - \left\{ \frac{1}{2.0} [\Delta g_Z(I,K) + \Delta g_Z(I,K+1)] \right\}}{TP(K+1) - TP(K)} \right] \quad (B-7)$$

and

$$V(I,J,K) = \left\{ [RL(I) - \Delta g_R(J,K)]^2 - [RL(I-1)]^2 \right\} [ZL(J) - ZL(J-1) - \Delta g_Z(I,K)] \\ \times [TL(K) - TL(K-1) - \Delta g_T(I,J)] (0.5). \quad (B-8)$$

For the rectangular geometry,

$$RR(I,J,K) = TL(K) - TL(K-1) - \frac{\Delta g_T(I,J) + \Delta g_T(I+1,J)}{2.0} \\ \times \frac{ZL(J) - ZL(J-1) - \left\{ \frac{1}{2.0} [\Delta g_Z(I,K) + \Delta g_Z(I+1,K)] \right\}}{RP(I+1) - RP(I)} \quad (B-9)$$

$$RZ(I,J,K) = TL(K) - TL(K-1) - \frac{\Delta g_T(I,J) + \Delta g_T(I,J+1)}{2.0} \\ \times \frac{RL(I) - RL(I-1) - \left\{ \frac{1}{2.0} [\Delta g_R(J,K) + \Delta g_R(J+1,K)] \right\}}{ZP(J+1) - ZP(J)} \quad (B-10)$$

$$\begin{aligned}
 RT(I, J, K) = & RL(I) - RL(I-1) - \frac{\Delta g_R(J, K) + \Delta g_R(J, K+1)}{2.0} \\
 & \times \frac{ZL(J) - ZL(J-1) - \left\{ \frac{1}{2.0} [\Delta g_Z(I, K) + \Delta g_Z(I, K+1)] \right\}}{TP(K+1) - TP(K)} \quad (B-11)
 \end{aligned}$$

and

$$\begin{aligned}
 V(I, J, K) = & [TL(K) - TL(K-1) - \Delta g_T(I, J)] [RL(I) - RL(I-1) - \Delta g_R(J, K)] \\
 & \times [ZL(J) - ZL(J-1) - \Delta g_Z(I, K)] \quad (B-12)
 \end{aligned}$$

APPENDIX C

COMPOSITE CONDUCTIVITIES

Depending upon the part of the TAC3D envelope being considered, heat is transported in the following ways:

- | | |
|-------------|--|
| 1. Material | Conduction |
| 2. Gap | Conduction (through gas) and
radiation (across gap) |
| 3. Coolant | Radiation (across coolant) and
convection (at coolant boundaries) |

For most thermal parameters (e.g., heat generation, specific heat) an average value around a mesh point is needed. However, for thermal conductivities, an average value between two mesh points is the required value. Because the data are supplied for mesh points, the average values must be calculated in each of the three coordinate directions. There are four basically different cases to be considered for each of the three directions and each of the two geometries. They are as follows:

1. No gap between adjacent mesh points, equal thermal conductivities
2. No gap between adjacent mesh points, unequal thermal conductivities
3. A gap between adjacent mesh points
4. Coolants

Figure C.1 illustrates the logic used by TAC3D in determining what calculations are to be performed.

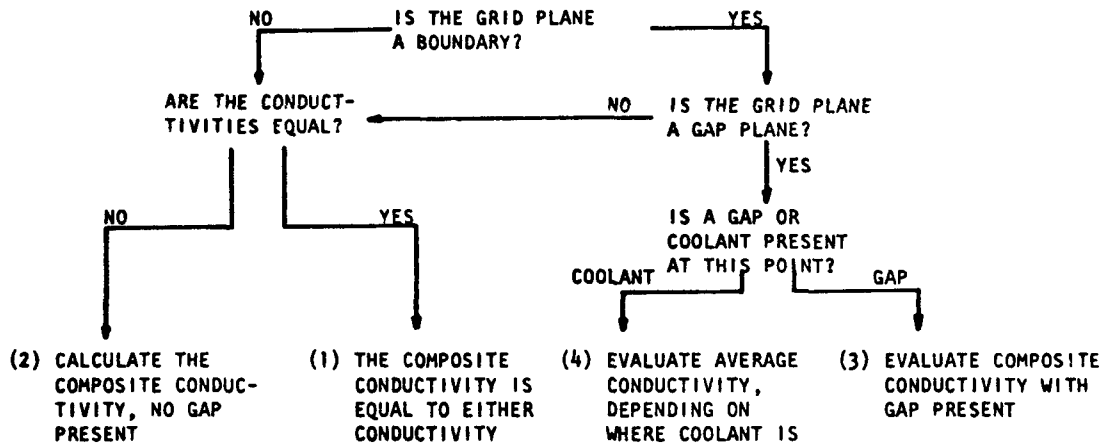


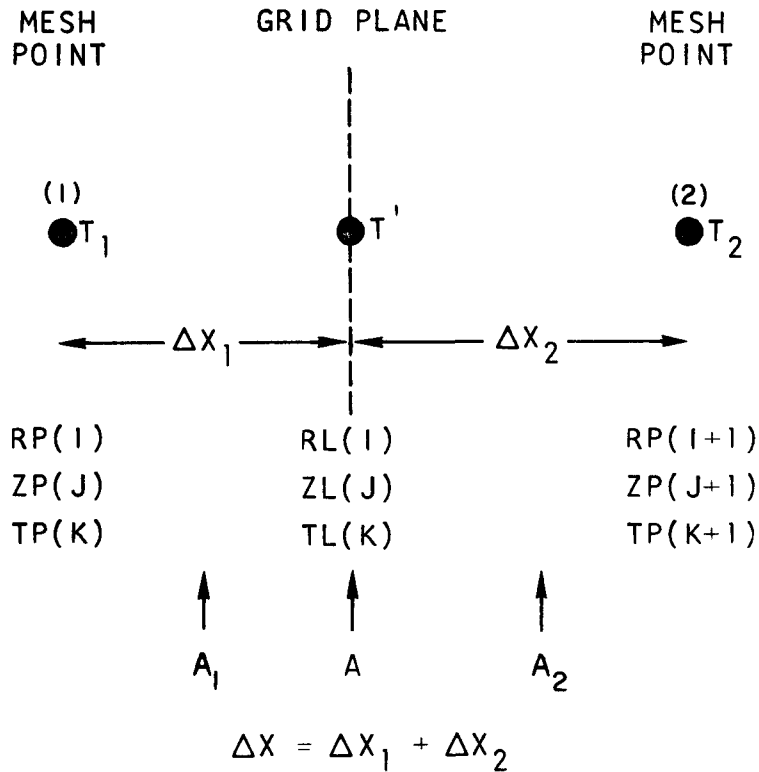
Fig. C.1. Logical search at each grid plane

C.1 NO GAP BETWEEN ADJACENT MESH POINTS, EQUAL THERMAL CONDUCTIVITIES

This case is trivial; the average thermal conductivity is equal to the thermal conductivity at either mesh point.

C.2 NO GAP BETWEEN ADJACENT MESH POINTS, UNEQUAL THERMAL CONDUCTIVITIES

The following sketch illustrates a condition where there is no gap between adjacent mesh points and thermal conductivities are unequal.



The thermal conductivities at mesh points 1 and 2, k_1 and k_2 are known. A composite thermal conductivity K between mesh points 1 and 2, is defined by

$$\frac{KA(T_1 - T_2)}{\Delta X} = \frac{k_1 A_1 (T_1 - T')}{\Delta X_1} = \frac{k_2 A_2 (T' - T_2)}{\Delta X_2} \quad , \quad (C-1)$$

where A is an average area across which heat is transferred, ΔX is a distance across which heat is transferred, and T' is the unit cell boundary temperature. Elimination of T' and resubstitution yields

$$K = \frac{k_1 k_2 \frac{\Delta X}{A}}{k_1 \frac{\Delta X_2}{A_2} + k_2 \frac{\Delta X_1}{A_1}} \quad . \quad (C-2)$$

The total resistance is the sum of the individual resistances. Therefore, for the geometry dependent components of these resistances,

$$\frac{\Delta X}{A} = \frac{\Delta X_2}{A_2} + \frac{\Delta X_1}{A_1} \quad (C-3)$$

Now, the equation for K may be rewritten as

$$K = \frac{1}{\frac{1}{k_2} + \left(\frac{1}{k_1} - \frac{1}{k_2} \right) \left(\frac{\Delta X_1/A_1}{\Delta X/A} \right)} \quad (C-4)$$

Considerations for the determination of the term $(\Delta X_1/A_1)/(\Delta X/A)$ in Eq. (C-4) are given below. In the radial direction for cylindrical geometry, per unit angle, per unit height, with the areas expressed as mean areas,

$$\frac{A}{\Delta X} = \frac{1}{\ln \{ [RP(I+1)]/[RP(I)] \}} \quad \text{and} \quad \frac{A_1}{\Delta X_1} = \frac{1}{\ln \{ [RL(I)]/[RP(I)] \}} \quad (C-5)$$

and, in the radial direction for rectangular geometry,

$$A = A_1 = A_2$$

$$\Delta X = RP(I+1) - RP(I) \quad (C-6)$$

$$\Delta X_1 = RL(I) - RP(I) \quad .$$

In the theta direction for cylindrical geometry,

$$A = A_1 = A_2$$

$$\Delta X = [TP(K+1) - TP(K)] [RP(I)] \quad (C-7)$$

$$\Delta X_1 = [TL(K) - TP(K)] [RP(I)] \quad .$$

and, in the theta direction for rectangular geometry,

$$A = A_1 = A_2$$

$$\Delta X = TP(K+1) - TP(K) \quad (C-8)$$

$$\Delta X_1 = TL(K) - TP(K) \quad .$$

In the axial direction for rectangular and cylindrical geometries,

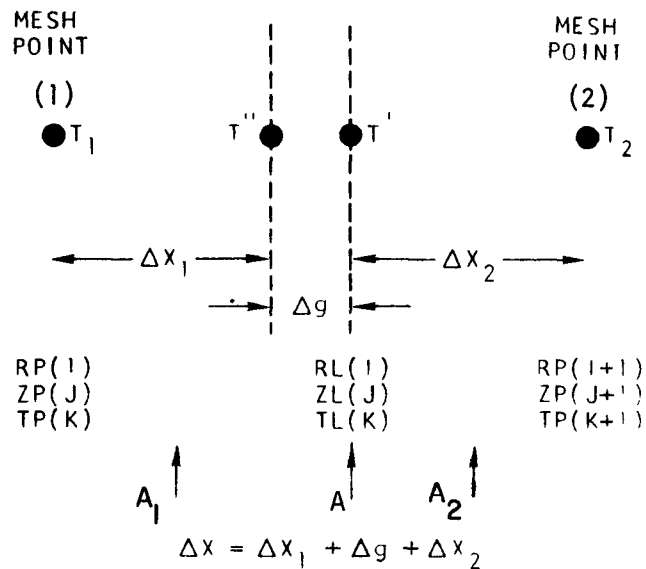
$$\begin{aligned}
 A &= A_1 = A_2 \\
 \Delta X &= ZP(J+1) - ZP(J) \\
 \Delta X_1 &= ZL(J) - ZP(J) \quad .
 \end{aligned}
 \tag{C-9}$$

The nomenclature and definition of the geometrical constant $(\Delta X_1/A_1)/(\Delta X/A)$ in the coordinate directions for the two available geometry types are given in Table C.1.

Table C.1
 GEOMETRICAL CONSTANT
 $(\Delta X_1/A_1)/(\Delta X/A)$

Name (Coordinate)	Definition	
	Cylindrical Geometry	Rectangular Geometry
RLN(I) (Radial)	$\frac{\ln\{[RL(I)]/[RP(I)]\}}{\ln\{[RP(I+1)]/RP(I)\}}$	$\frac{RL(I) - RP(I)}{RP(I+1) - RP(I)}$
ZLN(J) (Axial)	$\frac{ZL(J) - ZP(J)}{ZP(J+1) - ZP(J)}$	$\frac{ZL(J) - ZP(J)}{ZP(J+1) - ZP(J)}$
TLN(K) (Theta)	$\frac{TL(K) - TP(K)}{TP(K+1) - TP(K)}$	$\frac{TL(K) - TP(K)}{TP(K+1) - TP(K)}$

C.3 A GAP BETWEEN ADJACENT MESH POINTS



The thermal conductivities at mesh points 1 and 2 and in the gap are known. We define a composite thermal conductivity K between mesh points 1 and 2 as

$$\frac{KA(T_1 - T_2)}{\Delta X} \equiv \frac{k_1 A_1 (T_1 - T'')}{\Delta X_1} = \frac{k_2 A_2 (T' - T_2)}{\Delta X_2} = \frac{k_g A_g (T'' - T')}{\Delta g} + \frac{\sigma A_g (T''^4 - T'^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \quad (C-10)$$

where σ is the Stefan-Boltzmann constant ($.1713 \times 10^{-8}$ Btu/hr-ft²-°R⁴) and ϵ_1 and ϵ_2 are thermal emissivities. The surface temperatures of the gap, T' and T'' in Eq. (C-10), must be obtained by an iterative method.

C.3.1 Iterative Evaluation of Gap Surface Temperatures

If a value for T' is assumed, a better value may now be calculated by the algorithm

$$\frac{k_1 A_1}{\Delta X_1} (T_1 - T'') = \frac{k_2 A_2}{\Delta X_2} (T' - T_2) \quad (C-11)$$

When solving for T'' , one obtains

$$T'' = \frac{k_2 A_2 \Delta X_1}{k_1 A_1 \Delta X_2} (T_2 - T') + T_1 \equiv B(T_2 - T') + T_1 \quad (C-12)$$

$$\frac{k_2 A_2}{\Delta X_2} (T' - T_2) = \frac{k_g A_g}{\Delta g} (T'' - T') \left[1 + \frac{\sigma \Delta g (T''^2 + T'^2) (T'' + T')}{k_g \left(\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1 \right)} \right] \quad (C-13)$$

and

$$T' - T_2 = \frac{k_g A_g \Delta X_2}{k_2 A_2 \Delta g} (T'' - T') G \equiv H(T'' - T') \quad (C-14)$$

When substituting for T'' and solving for T' , one obtains

$$T' - T_2 = H[B(T_2 - T') + T_1 - T'] \quad , \quad (C-15)$$

and

$$T' = \frac{T_2(HB + 1) + HT_1}{HB + 1 + H} \quad . \quad (C-16)$$

If one uses this algorithm repeatedly, T' may be approximated to the desired accuracy.

Because H and B and the temperatures are always positive, T' must lie between T_1 and T_2 , regardless of what initial temperature was guessed for T' . No formal proof of convergence is offered, but, in practice, this algorithm has converged rapidly in every case in which it was used.

This treatment contains two inaccuracies. The first inaccuracy is introduced because the average area for heat transfer across the gap is not always the same for radiation and conduction. In TAC3D, this area is determined at the outside of the gap. In other words, if in the preceding figure the coordinate X increases to the right, then A_g is assumed to be the same as A . The other inaccuracy is introduced in the iterative method of obtaining T' ; however, this error can be made as small as desired. Presently, the iterations are stopped when two consecutive values of T' are within 1°F of each other. Up to ten iterations are allowed to accomplish this.

If either T_1 or T_2 is negative, the convergence scheme diverges in three or four iterations. An error return stops the problem at this time.

C.3.2 Geometrical Constants for Gaps

Three geometrical terms -- $(A_2/\Delta X_2)/(A/\Delta X)$, $(A_2/\Delta X_2)/(A_1/\Delta X_1)$, and $A_g/(A_2/\Delta X_2)$ -- were required for the equations that determine

gap surface temperatures and composite conductivity with a gap present. Considerations for the determination of these terms are given below.

In the radial direction for cylindrical geometry per unit angle, per unit height and with the areas expressed as mean areas,

$$A/(\Delta X) = \frac{1}{\ln\{[RP(I + 1)]/[RP(I)]\}} \quad ,$$

$$A_2/(\Delta X_2) = \frac{1}{\ln\{[RP(I + 1)]/[RL(I)]\}} \quad ,$$

$$A_1/(\Delta X_1) = \frac{1}{\ln\{[RL(I) - \Delta g_R]/[R(I)]\}} \quad ,$$

$$A_g = RL(I) \quad (\text{per unit arc}) \quad ,$$

and

$$\Delta g = \Delta g_R \quad . \quad (C-17)$$

In the radial direction for rectangular geometry,

$$A = A_1 = A_2 = A_g \quad ,$$

$$\Delta X = RP(I + 1) - RP(I) \quad ,$$

$$\Delta X_1 = RL(I) - RP(I) - \Delta g_R \quad ,$$

$$\Delta X_2 = RP(I + 1) - RL(I) \quad ,$$

and

$$\Delta g = \Delta g_R \quad . \quad (C-18)$$

In the axial direction for rectangular and cylindrical geometries,

$$A = A_1 = A_2 = A_g \quad ,$$

$$\Delta X = ZP(J + 1) - ZP(J) \quad ,$$

$$\begin{aligned}\Delta X_1 &= ZL(J) - ZP(J) - \Delta g_Z \quad , \\ \Delta X_2 &= ZP(J + 1) - ZL(J) \quad ,\end{aligned}\tag{C-19}$$

and

$$\Delta g = \Delta g_Z \quad .$$

In the theta direction for cylindrical geometry,

$$\begin{aligned}A &= A_1 = A_2 = A_g \quad , \\ \Delta X &= [TP(K + 1) - TP(K)][RP(I)] \quad , \\ \Delta X_1 &= [TL(K) - TP(K) - \Delta g_Z][RP(I)] \quad , \\ \Delta X_2 &= [TP(K + 1) - TL(K)][RP(I)] \quad ,\end{aligned}$$

and

$$\Delta g = [RP(I)](\Delta g_Z)\tag{C-20}$$

In the theta direction for rectangular geometry,

$$\begin{aligned}A &= A_1 = A_2 = A_g \quad , \\ \Delta X &= TP(K + 1) - TP(K) \quad , \\ \Delta X_1 &= TL(K) - TP(K) - \Delta g_T \quad , \\ \Delta X_2 &= TP(K + 1) - TL(K) \quad ,\end{aligned}$$

and

$$\Delta g = \Delta g_T \quad .\tag{C-21}$$

The nomenclature and definition of the geometrical constants $(A_2/\Delta X_2)/(A/\Delta X)$, $(A_2/\Delta X_2)/(A_1/\Delta X_1)$, and $A_g/(A_2/\Delta X_2)$ in the coordinate directions for the two available geometry types are given in Tables C.2, C.3, and C.4.

Table C.2
 GEOMETRICAL CONSTANT
 $(A_2/\Delta X_2)/(A/\Delta X)$

Name (Gap Plane)	Grid Plane	Definition	
		Cylindrical	Rectangular
RATIOK(IG)	I	$\frac{\ln\{[RP(I+1)]/[RP(I)]\}}{\ln\{[RP(I+1)]/[RL(I)]\}}$	$\frac{RP(I+1)-RP(I)}{RP(I+1)-RL(I)}$
ZATIOK(JG)	J	$\frac{ZP(J+1)-ZP(J)}{ZP(J+1)-ZL(J)}$	$\frac{ZP(J+1)-ZP(J)}{ZP(J+1)-ZL(J)}$
TATIOK(KG)	K	$\frac{TP(K+1)-TP(K)}{TP(K+1)-TL(K)}$	$\frac{TP(K+1)-TP(K)}{TP(K+1)-TL(K)}$

Table C.3
 GEOMETRICAL CONSTANT
 $(A_2/\Delta X_2)/(A_1/\Delta X_1)$

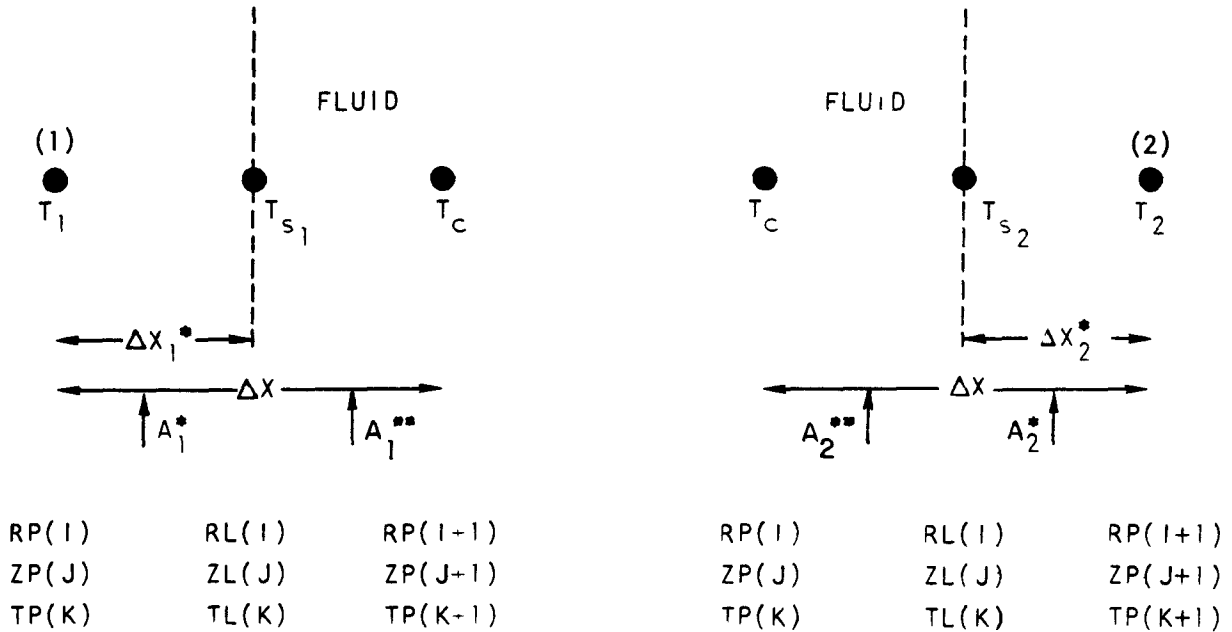
Name (Gap Plane)	Point Limits	Definition	
RATIOB(IG, J, K)	J=JLS-JHS K=KLS-KHS Block L	Cylindrical	$\ln\left[\frac{RL(IHS)-RDG(L)}{RP(IHS)}\right] / \ln\left[\frac{RP(IHS+1)}{RL(IHS)}\right]$
		Rectangular	$\frac{RL(IHS)-RP(IHS)-RDG(L)}{RP(IHS+1)-RL(IHS)}$
ZATIOB(JG, I, K)	I=ILS-IHS K=KLS-KHS Block L	Cylindrical	$\frac{ZL(JHS)-ZP(JHS)-ZDG(L)}{ZP(JHS+1)-ZL(JHS)}$
		Rectangular	$\frac{ZL(JHS)-ZP(JHS)-ZDG(L)}{ZP(JHS+1)-ZL(JHS)}$
TATIOB(KG, I, J)	I=ILS-IHS J=JLS-JHS Block L	Cylindrical	$\frac{TL(KHS)-TP(KHS)-TDG(L)}{TP(KHS+1)-TL(KHS)}$
		Rectangular	$\frac{TL(KHS)-TP(KHS)-TDG(L)}{TP(KHS+1)-TL(KHS)}$

Table C.4
 GEOMETRICAL CONSTANT
 $A_g/(A_2/\Delta X_2)$

Name (Gap Plane)	Grid Plane	Definition	
		Cylindrical	Rectangular
RATIOH(IG)	I	$RL(I) \left[\ln \frac{RP(I+1)}{RL(I)} \right]$	$RP(I+1)-RL(I)$
ZATIOH(JG)	J	$ZP(J+1)-ZL(J)$	$ZP(J+1)-ZL(J)$
TATIOH(KG)	K	$TP(K+1)-TL(K)$	$TP(K+1)-TL(K)$

C.4 COOLANTS

In the calculation of the composite conductivity for points bordering a coolant, one of the following illustrations applies, depending upon the point being considered.



T_{s1}, T_{s2} = SURFACE TEMPERATURES T_c = COOLANT TEMPERATURE
--

Then, depending upon relative coolant placement, the three cases present are:

1. Coolant on high index side of grid plane
2. Coolant on low index side of grid plane
3. Two coolants adjacent

C.4.1 Coolant on High Index Side of Grid Plane

When one knows the thermal conductivity k_1 at mesh point 1 and the heat transfer coefficient at the coolant surface h_1 , the composite thermal conductivity K_1 between the mesh point and the fluid is

$$\frac{k_1 A_1^*}{\Delta X_1^*} (T_1 - T_{s_1}) = h_1 A_1 (T_{s_1} - T_c) + \frac{\sigma A_1 (T_{s_1}^4 - T_{s_2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1}$$

$$= \frac{K_1 A_1^{**}}{\Delta X} (T_1 - T_c) \quad , \quad (C-22)$$

where

A_1 = area across which heat is transferred at the coolant surfaces

A_1^* = average area across which heat is transferred in the material

A_1^{**} = average area for heat transfer between T_1 and T_c

ΔX_1^* = distance across which heat travels to reach the coolant surface

σ = Stefan-Boltzmann constant, 0.1713×10^{-8} Btu/hr-ft²-°R⁴

ϵ_1, ϵ_2 = thermal emissivities.

The term $h_1 A_1 (T_{s_1} - T_c)$ represents the heat transfer by convection to the fluid, and

$$\frac{\sigma A_1 (T_{s_1}^4 - T_{s_2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \quad (C-23)$$

gives the value of the heat transfer by radiation to the opposite surface.

A composite heat transfer coefficient H_1 is defined by

$$h_1 A_1 (T_{s_1} - T_c) + \frac{\sigma A_1 (T_{s_1}^4 - T_{s_2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} = H_1 A_1 (T_{s_1} - T_c). \quad (C-24)$$

Let

$$\frac{\sigma (T_{s_1}^4 - T_{s_2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} = \text{RAD} \quad . \quad (C-25)$$

Now, the surface temperature T_{s_1} is calculated as

$$\frac{k_1 A_1^*}{\Delta X_1} (T_1 - T_{s_1}) = H_1 A_1 (T_{s_1} - T_c) \quad , \quad (C-26)$$

and

$$T_{s_1} = \frac{H_1 A_1 T_c + \frac{k_1 A_1^*}{\Delta X_1} T_1}{H_1 A_1 + \frac{k_1 A_1^*}{\Delta X_1}} = \frac{\frac{1}{k_1} T_c + \frac{1}{H_1} \frac{A_1^*/\Delta X_1}{A_1} T_1}{\frac{1}{k_1} + \frac{1}{H_1} \frac{A_1^*/\Delta X_1}{A_1}} \quad . \quad (C-27)$$

By substitution of T_{s_1} in Eq. (C-24) one may solve for H_1 as

$$\frac{1}{H_1} = \frac{T_1 - T_c - \frac{\text{RAD}}{k_1} \frac{A_1}{A_1^*/\Delta X_1}}{h_1 \left(T_1 - T_c + \frac{\text{RAD}}{h_1} \right)} \quad . \quad (C-28)$$

Rewriting (C-22) as

$$\frac{K_1 A_1^{**}}{\Delta X} (T_1 - T_c) = H_1 A_1 (T_{s_1} - T_c) \quad , \quad (C-29)$$

one may solve for K_1 by substitution of (C-27) in (C-29):

$$K_1 = \frac{1}{\frac{1}{k_1} \frac{A_1^{**}}{\Delta X_1^*} + \frac{1}{H_1} \frac{A_1^{**}}{A_1}} \quad (C-30)$$

C.4.2 Coolant on Low Index Side of Grid Plane

When one knows the thermal conductivity k_2 at mesh Point 2 and the heat transfer coefficient at the coolant surface h_2 , the composite thermal conductivity K_2 between the mesh point and the fluid is

$$\begin{aligned} \frac{k_2 A_2^*}{\Delta X_2^*} (T_2 - T_{s_2}) &= h_2 A_2 (T_{s_2} - T_c) + \frac{\sigma A_1 (T_{s_2}^4 - T_{s_1}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \\ &= \frac{K_2 A_2^{**} (T_2 - T_c)}{\Delta X} \end{aligned} \quad (C-31)$$

where

A_2 = area across which heat is transferred at the coolant surface

A_2^* = average area across which heat is transferred in the solid

A_2^{**} = average area for heat transfer between T_2 and T_c

ΔX_2^* = distance across which heat travels to reach the coolant surface

A composite heat transfer coefficient H_2 is defined by

$$h_2 A_2 (T_{s_2} - T_c) - \frac{\sigma A_1 (T_{s_1}^4 - T_{s_2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} = H_2 A_2 (T_{s_2} - T_c) \quad (C-32)$$

When one follows the same scheme as with the coolant on the high index side of the grid plane, the surface temperature T_{s_2} , the composite

heat transfer coefficient H_2 and the composite thermal conductivity K_2 are obtained as

$$T_{s2} = \frac{\frac{1}{k_2} T_c + \frac{1}{H_2} \frac{A_2^*/\Delta X_2^*}{A_2} T_2}{\frac{1}{k_2} + \frac{1}{H_2} \frac{A_2^*/\Delta X_2^*}{A_2}}, \quad (C-33)$$

and

$$\frac{1}{H_2} = \frac{T_2 - T_c + \frac{RAD}{k_2} \frac{A_1}{A_2} \frac{A_2}{A_2^*/\Delta X_2^*}}{h_2 \left(T_2 - T_c - \frac{RAD}{h_2} \frac{A_1}{A_2} \right)} \quad (C-34)$$

and

$$K_2 = \frac{1}{\frac{1}{k_2} \frac{A_2^{**}/\Delta X}{A_2^*/\Delta X_2^*} + \frac{1}{H_2} \frac{A_2^{**}/\Delta X}{A_2}} \quad (C-35)$$

C.4.3 Geometrical Constants for Coolants

Some of the geometrical constants appearing in Eqs. (C-27) through (C-35) are given below. The nomenclature used is that previously defined in Appendix B and in Tables C.1 and C.2 of this appendix.

For $(A_1^*/\Delta X_1^*)/A_1$,

$$\text{Radial} = \frac{1.0}{[\text{RP}(I + 1) - \text{RP}(I)] \text{RLN}(I)}$$

$$\text{Axial} = \frac{1.0}{\text{ZL}(J) - \text{ZP}(J)}$$

$$\text{Theta} = \frac{1.0}{[\text{TL}(K) - \text{TP}(K)] [\text{RP}(I)]} \quad (C-36)$$

For $(A_2^*/\Delta X_1^*)/A_1$,

$$\begin{aligned} \text{Radial} &= \frac{\text{RATIOK(IG)}}{\text{RP(I + 1)} - \text{RP(I)}} \\ \text{Axial} &= \frac{1.0}{\text{ZP(J + 1)} - \text{ZL(J)}} \\ \text{Theta} &= \frac{1.0}{[\text{TP(K + 1)} - \text{TL(K)}][\text{RP(I)}]} \end{aligned} \quad (\text{C-37})$$

For $(A_1^{**}/\Delta X)/A_1$ and $(A_2^{**}/\Delta X)/A_2$,

$$\begin{aligned} \text{Radial} &= \frac{1.0}{\text{RL(I)} \times \ln \left[\frac{\text{RP(I + 1)}}{\text{RP(I)}} \right]} \\ \text{Axial} &= \frac{1.0}{\text{ZP(J + 1)} - \text{ZP(J)}} \\ \text{Theta} &= \frac{1.0}{[\text{TP(K + 1)} - \text{TP(K)}][\text{RP(I)}]} \end{aligned} \quad (\text{C-38})$$

For A_1/A_2 ,

$$\begin{aligned} \text{Radial} &= \frac{\text{RL(ILS - 1)}}{\text{RL(IHS)}} \\ \text{Axial} &= 1.0 \\ \text{Theta} &= 1.0 \end{aligned} \quad (\text{C-39})$$

In the above, IHS and ILS are the indices, respectively, of the highest and lowest radial point levels in the coolant block.

C.4.4 Two Coolants Adjacent

In this case, no heat transfer is allowed between the two adjacent coolants.

APPENDIX D

COOLANTS

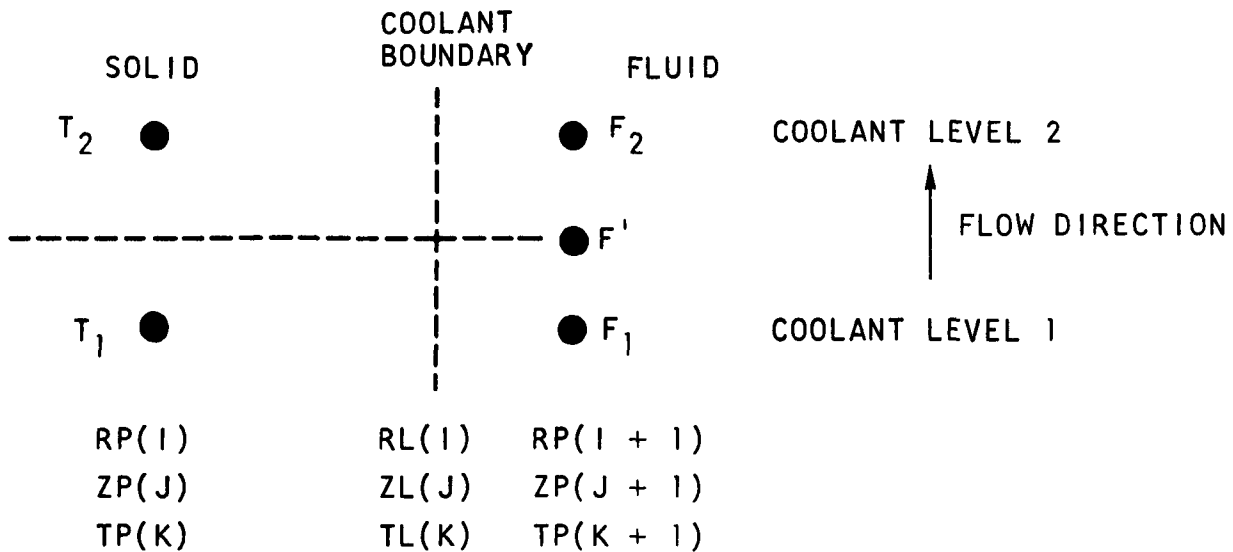
The calculation of composite heat transfer coefficients for mesh points adjacent to or inside coolants was considered in Appendix C. In this appendix, the coolant temperature calculation is discussed. With respect to the TAC3D code, the following scheme is used in handling the coolants:

1. The solution for mesh point temperatures is carried out as described in Appendix A. During this solution, the points in coolants experience no change in temperature. This is accomplished by setting the thermal capacitance of all coolant nodes at 1.0×10^{12} Btu/°R. The effect of radiation and convection at coolant boundaries on mesh point temperatures adjacent coolants is accounted for in the solution through use of the composite conductivities described in Section C.4 of Appendix C.
2. New coolant temperatures based on the results of the above solution are calculated.
3. New coolant block boundary temperatures based on the results of Steps 2 and 3 above are calculated. These are used to determine a new value for the radiation transport as defined in Eq. (C-25) of Appendix C.
4. Composite thermal conductivities between the bounding material and the coolant (Eqs. (C-30) and (C-35) of Appendix C) are calculated, and another iteration is performed.

In TAC3D, the assumption is made that the coolant is, at any given time, in equilibrium with its surface. This implies that the coolant passes by the surface in a time considerably shorter than the smallest time step used. This means that the code should not be used to study

transient behavior of the coolant over time intervals comparable to coolant transit time.

A coolant boundary along the flow direction is illustrated in the following figure.



FLOW RATE	W (LB/HR)
SPECIFIC HEAT	C_p (BTU/LB-°R)
COOLANT TEMPERATURE	F (°R)
SOLID TEMPERATURE	T (°R)

It is assumed that there are no temperature gradients in the coolant normal to its flow direction. Therefore the differential heat balance for the coolant surface may be stated as

$$WC_p dF = \sum_i h_i S_i (T_i - F) dz \quad , \quad (D-1)$$

where

- dF = differential coolant temperature increment
- h_i = heat transfer coefficient (Btu/ft²-hr-°R)
- S_i = linear surface along coolant boundary associated with one point
- dz = differential height of increment along flow direction
- T_i = surface temperature

Rearrangement gives

$$dF = \left(\sum_i h_i S_i T_i - \sum_i h_i S_i F \right) \frac{dx}{WC_p} \quad (D-2)$$

When

$$M = \sum_i h_i S_i T_i \quad \text{and} \quad H = \sum_i h_i S_i$$

are defined, the equation becomes

$$dF = (M - HF) \frac{dz}{WC_p} \quad \text{or} \quad \frac{dF}{M - HF} = \frac{dz}{WC_p} \quad (D-3)$$

Now, integrating from one grid line (B) to the next one (T), we get

$$\ln(M_T - H_T F_T) - \ln(M_B - H_B F_B) = -\frac{H_T Z_T}{WC_p} + \frac{H_B Z_B}{WC_p} \quad (D-4)$$

Because $M_T + M_B$, $H_T = H_B$, and $Z = Z_T - Z_B$ (the finite increment of height),

$$\ln \left[\frac{M - HF_T}{M - HF_B} \right] = -\frac{HZ}{WC_p},$$

$$M - HF_T = (M - HF_B) e^{-HZ/WC_p},$$

and

$$F_T = \frac{M}{H} - \left(\frac{M}{H} - F_B \right) e^{-HZ/WC_p} \quad (D-5)$$

Referring to the definitions of M and H and multiplying by the increment of height, we get

$$\frac{M}{H} = \frac{MZ}{HZ} = \frac{\sum_i h_i S_i T_i Z}{\sum_i h_i S_i Z} = \frac{\sum_i h_i A_i T_i}{\sum_i h_i A_i} \quad (D-6)$$

In deriving the previous equations, the T_i 's were surface temperatures and the S_i 's were surface areas.

In Appendix C, Section C.4, the composite thermal conductivities, K_i , between mesh points in materials and adjacent mesh points in coolants are derived. The composite overall thermal conductances between such sets of points are

$$C_i = \left(\frac{A}{\Delta X} \right)_i K_i \quad (D-7)$$

where the $(A/\Delta X)$ terms are the geometry factors given in Appendix B. If the T_i are now redefined to be the temperatures of the mesh points adjacent to the coolant, then C_i corresponds to $h_i A_i$ and Eqs. (D-6) and (D-5) may be restated as

$$\frac{M}{H} = \frac{\sum_i C_i T_i}{\sum_i C_i} \quad (D-8)$$

$$F_T = \frac{\sum_i C_i T_i}{\sum_i C_i} \left(1 - e^{-\sum_i C_i / WC_p} \right) + F_B e^{-\sum_i C_i / WC_p} \quad (D-9)$$

When

$$P = \sum_i C_i T_i \text{ and } R = \sum_i C_i, \quad (D-10)$$

$$F_T = \frac{P}{R} \left(1 - e^{-R/WC_p} \right) + F_B e^{-R/WC_p} .$$

The average coolant temperature along the increment F_{av} is defined by a heat balance as

$$\begin{aligned} WC_p (F_T - F_B) &= \sum_i C_i (T_i - F_{av}) \\ &= \sum_i C_i T_i - \sum_i C_i F_{av} \\ &= P - R F_{av}, \end{aligned} \quad (D-11)$$

and

$$F_{av} = \frac{P - WC_p (F_T - F_B)}{R} \quad (D-12)$$

As one progresses along the direction of the coolant flow, in steps, F_T and F_{av} are computed for each level. The F_B for a level is the F_T of the previous level. The F_T of the final level is the outlet temperature for the coolant.



APPENDIX E

STEADY STATE OPTION

If only the steady-state solution is of interest, the system of equations may be solved according to the method given in Appendix A using arbitrary values for the material specific heats and time steps. This pseudo-transient calculation is extended to the point where results are no longer time dependent. The efficiency of such a calculation depends on the ability of the code user to choose time steps which are large, yet not so large as to cause the solution to become unstable. Generally, the user must approximate the value of such time steps by trial and error. Furthermore, the stability characteristics of mesh points vary throughout any given problem so that the maximum time step which may be used is often governed by the behavior of only a few points. TAC3D contains a steady-state option which, for most problems, has been found to circumvent these difficulties when only a steady-state solution is desired. All calculations are performed in the normal manner except that control of the pseudo-transient is transferred to a special subroutine which assigns the capacitance terms and time steps by the method described below.

For explicit methods of solution, Ref. 5 gives a criterion for the maximum stable time step at a mesh point.

$$\Delta\tau_i \leq \frac{C_i}{\sum_j K_{ji}} \quad (E-1)$$

where

$\Delta\tau_i$ is the maximum stable time step for point i , hr

C_i is the thermal capacitance associated with point i , Btu/°R

K_{ji} is the thermal conductance between point i and a neighboring point j , Btu/hr-°R

For an implicit method of solution such as that used in TAC3D, $\Delta\tau_i$ has been found through experience to be proportional to the right hand side of Eq. (E-1). Therefore, at the limit of stability

$$\Delta\tau_i = \beta_i \frac{C_i}{\sum_j K_{ji}} \quad (\text{E-2})$$

where the β_i are unknown constants of proportionality.

In TAC3D, Eq. (E-2) is used to make the $\Delta\tau_i$ approximately equal to a single value $\Delta\tau'$ for all mesh points. Assuming

$$\Delta\tau' = 1.0 \text{ hr}$$

and

$$\text{all } \beta_i = 1.0,$$

Eq. (E-2) becomes

$$C_i = \sum_{j=1}^6 K_{ji} \quad (\text{E-3})$$

Eq. (E-3) is used to assign the thermal capacitance at each mesh point. The result is that all points will have roughly similar stability characteristics with respect to a time step $\Delta\tau'$. The effectiveness of this treatment depends upon the validity of the assumption of uniform β_i which can vary depending on the nature of the problem. Because the K_{ji} may be temperature dependent, the C_i are recalculated for each iteration. However, in coolants and in dummy materials which simulate either constant temperature source-sinks or near perfect insulators, the C_i are set at 1.0×10^{12} Btu/ $^{\circ}$ R and never changed. In coolants (see Appendix D) this action is taken whether or not the steady-state option is being used.

The actual value of the $\Delta\tau'$ to be used for each iteration is found by trial and error. Starting with an initial value of one hour it is increased up to a maximum value of three hours according to the function

$$\Delta\tau' = \frac{N}{10} + 1 \quad (\text{E-4})$$

N is the number of iterations which have been previously completed. The results of each iteration are monitored for indications of gross instability. If any point temperature T_i falls outside the range

$$0.0 < T_i < 1.0 \times 10^6 \text{ } ^\circ\text{R}$$

then the current value of $\Delta\tau$ is divided by 1.50 and no further increase is allowed. Also the current results are replaced by older results (ten to twenty iterations older) which have been retained on an external storage device. This is done to prevent the unstable condition just detected from being propagated into the solution for the next iteration.

As soon as the problem is converged, a series of twenty smoothing iterations is performed. The C_i retain their values from the last steady-state iteration while the time step is reduced by a factor of ten and held constant. The purpose of the smoothing iterations is to eliminate any small oscillations which may be present at the time convergence is apparently attained.

The degree to which the problem has converged upon its steady-state solution is assessed by checking the residuals, R_i , which are defined as

$$R_i = \frac{C_i (T_i - T'_i)}{\Delta\tau'_i} \quad (E-5)$$

$$\sum_{j=1}^6 \left| K_{ji} \left(\frac{(T_i + T'_i)}{2} - \frac{(T_j + T'_j)}{2} \right) \right| + |W_i|$$

where W_i is the heat generation rate in Btu/hr and all other symbols are as previously defined in this appendix. The primed values of T indicate point temperature results of the preceding iteration.* The R_i are calculated after every tenth iteration. The convergence criterion which must be satisfied is

$$\text{all } R_i < 1.0 \times 10^{-4} .$$

R_i values at points where, as previously discussed, the C_i have been set at 1.0×10^{12} Btu/ $^\circ\text{R}$ are not meaningful and are therefore not included in testing for convergence.

*These values are retained on an external storage device.



APPENDIX F

SAMPLE PROBLEM

A sample problem is presented to compare the results from TAC3D with an analytical solution. The objective is to show that the TAC3D program yields the correct solution.

F.1 PROBLEM DESCRIPTION

The sample problem requires evaluating transient temperature distributions in a rectangular parallelepiped of dimensions a , b , and c . The parallelepiped is composed of one material having constant properties and no internal heat generation. The faces at $x = 0$, $y = 0$, and $z = 0$ are adiabatic. The $x = a$ and $y = b$ faces have infinite heat transfer coefficients, and the $x = c$ face has a finite, non-zero heat transfer coefficient. The initial temperature throughout the parallelepiped is T_i . At time zero, the non-adiabatic faces are subjected to a fluid sink at T_f .

Mathematically, the problem is described as

$$0 < x < a; 0 < y < b; 0 < z < c$$

$$T(x,y,z,0) = T_i$$

$$\frac{\partial T}{\partial x}(0,y,z,t) = 0; \quad T(a,y,z,t) = T_f$$

$$\frac{\partial T}{\partial x}(x,0,z,t) = 0; \quad T(x,b,z,t) = T_f$$

$$\frac{\partial T}{\partial z}(x,y,0,t) = 0; \quad -k \frac{\partial T}{\partial z}(x,y,c,t) = h[T(x,y,c,t) - T_f]$$

For the specific problem analyzed, the following numerical values were used:

$$a = 4.2 \text{ in.}; b = 3.6 \text{ in.}; c = 2.4 \text{ in.}$$

$$k = 10 \text{ Btu/hr-ft-}^\circ\text{F}; \quad \rho c_p = 50 \text{ Btu/ft}^3\text{-}^\circ\text{F}$$

$$h = 50 \text{ Btu/hr-ft}^2\text{-}^\circ\text{F}$$

$$T_i = 600^\circ\text{F}; T_f = 100^\circ\text{F}$$

F.2 ANALYTICAL SOLUTION

The analytical solution, constructed from partial solutions given in Ref. 6, is

$$\frac{T - T_f}{T_i - T_f} = X(x,t) Y(y,t) Z(z,t)$$

where

$$X(x,t) = \frac{4}{\pi} \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)} \cos\left[\frac{(2n+1)\pi x}{2a}\right] e^{-\left[\frac{\kappa(2n+1)^2\pi^2 t}{4a^2}\right]}$$

$$Y(y,t) = \frac{4}{\pi} \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m+1)} \cos\left[\frac{(2m+1)\pi y}{2b}\right] e^{-\left[\frac{\kappa(2m+1)^2\pi^2 t}{4b^2}\right]}$$

$$Z(z,t) = \sum_{\ell=1}^{\infty} \frac{2(hc/k) \cos(\alpha_\ell z)}{\left\{\left[(hc/k)^2 + (\alpha_\ell c)^2\right] + hc/k\right\}} e^{-\kappa\alpha_\ell^2 t}$$

In the above, the thermal diffusivity κ is $\rho c_p/k$ and the eigenvalues α_ℓ are the roots of $\alpha_\ell c \tan(\alpha_\ell c) = hc/k$.

In obtaining numerical values for the analytical solution, two hundred terms were used in each of the series expansions, which were found to be more than sufficient. The first nine eigenvalues of the equation $\alpha_\ell c \tan(\alpha_\ell c) = hc/k$ were obtained from Ref. 7. With $hc/k = 1.0$, the asymptotic expression $\alpha_\ell = (\ell - 1)\pi$ was used for the higher eigenvalues. This expression was found to be accurate within .15%.

F.3 NUMERICAL SOLUTION WITH TAC3D

The model for the sample problem is shown in Fig. F.1. The solid material is represented by one material block. The boundary conditions on the six faces are described by coolants 1 through 6 (specifically, coolant blocks 1 through 6). The material and coolant thermal parameters

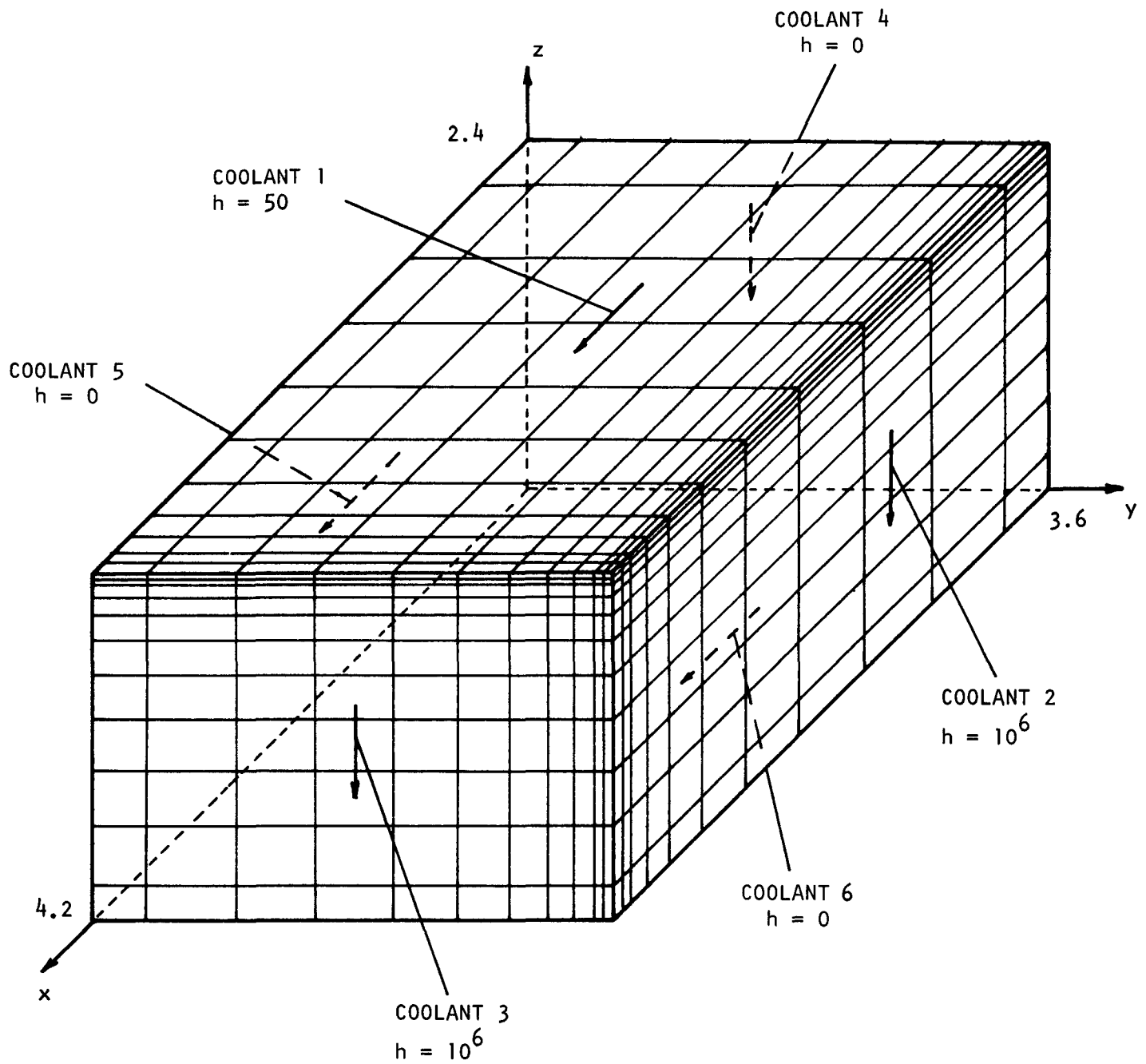


Fig. F.1. TAC3D sample problem — thermal model

as specified by the input functions, are given in Table F.1. All coolants have the standard flow rate of $10^6 \text{ lb}_m/\text{hr}$ and the standard specific heat of $1.0 \text{ Btu/lb}_m\text{-}^\circ\text{F}$. Coolants 4, 5, and 6, which represent the adiabatic faces, have the standard heat transfer coefficient equal to $10^{-6} \text{ Btu/hr-ft}^2\text{-}^\circ\text{F}$. A portion of the TAC3D printout information, Table F.2, shows the block and grid plane specifications, as well as time history information to run to 0.03 hrs.

F.4 COMPARISON OF THE TAC3D SOLUTION WITH THE ANALYTICAL SOLUTION

A comparison of the TAC3D solution with the analytical solution is given in Fig. F.2, where the temperature profile along the main diagonal is plotted at several times. This solution can also be checked with transient heat transfer charts such as those given in Ref. 8.

TABLE F.1
INPUT THERMAL PARAMETER FUNCTIONS
FOR TAC3D SAMPLE PROBLEM

C MATERIAL THERMAL PARAMETERS

RCON1(X)=10.

ACON1(X)=10.

TCON1(X)=10.

SPEC1(X)=50.

C COOLANT THERMAL PARAMETERS

H1A(X)=50.

TIN1A(X)=560.

H2A(X)=1.0E+6

TIN2A(X)=560.

H3A(X)=1.0E+6

TIN3A(X)=560.

TABLE F.2

EXCERPTS FROM TAC3D SAMPLE PROBLEM OUTPUT SHOWING
BLOCK, GRID PLANE AND TIME HISTORY SPECIFICATIONS

BLOCK NUMBER	BOUNDARIES						GAPS			
	LOW X (INCHES)	HIGH X (INCHES)	LOW Y (INCHES)	HIGH Y (INCHES)	LOW Z (INCHES)	HIGH Z (INCHES)	MATERIAL	X MATERIAL (INCHES)	Y MATERIAL (INCHES)	Z MATERIAL (INCHES)
1	.0000	4.2000	.0000	3.6000	.0000	2.4000	1			
2	.0000	4.2000	.0000	3.6000	2.4000	2.4000	-1			
3	.0000	4.2000	3.6000	3.6000	.0000	2.4000	-2			
4	4.2000	4.2000	.0000	3.6000	.0000	2.4000	-3			
5	.0000	.0000	.0000	3.6000	.0000	2.4000	-4			
6	.0000	4.2000	.0000	.0000	.0000	2.4000	-5			
7	.0000	4.2000	.0000	3.6000	.0000	.0000	-6			

TABLE F.2
(Continued)

SEQUENCE NUMBER		X BOUNDARY ASSIGNMENTS		
		POINT LOCATION (INCHES)	GRID LINE LOCATION (INCHES)	COOLANT OR GAP BOUNDARY NUMBER
1		.0000		
2	1	.4200	.0000	1
	2		.8400	
3		1.1550		
4	3	1.7850	1.4700	
5	4	2.4150	2.1000	
6	5	2.9400	2.7300	
7	6	3.3600	3.1500	
8	7	3.6750	3.5700	
9	8	3.8850	3.7800	
10	9	4.0530	3.9900	
11	10	4.1580	4.1160	
12	11	4.2000	4.2000	2

TABLE F.2
(Continued)

Y BOUNDARY ASSIGNMENTS			
SEQUENCE NUMBER	POINT LOCATION (INCHES)	GRID LINE LOCATION (INCHES)	COOLANT OR GAP BOUNDARY NUMBER
1	.0000		
2	.3600	.0000	1
3	.9900	.7200	
4	1.5300	1.2600	
5	2.0700	1.8000	
6	2.5200	2.3400	
7	2.8800	2.7000	
8	3.1500	3.0600	
9	3.3300	3.2400	
10	3.4740	3.4200	
11	3.5640	3.5280	
12	3.6000	3.6000	2

TABLE F.2
(Continued)

Z BOUNDARY ASSIGNMENTS			
SEQUENCE NUMBER	POINT LOCATION (INCHES)	GRID LINE LOCATION (INCHES)	COOLANT OR GAP BOUNDARY NUMBER
1	.0000		
2	.2400	.0000	1
3	.6000	.4000	
4	1.0200	.8400	
5	1.3000	1.2000	
6	1.6000	1.5000	
7	1.9200	1.8000	
8	2.1000	2.0400	
9	2.2200	2.1000	
10	2.3160	2.2000	
11	2.3760	2.3520	
12	2.4000	2.4000	2

TIME HISTORY

END OF TIME PERIOD			TIME STEP			PRINT FREQUENCY
HOURS	MINUTES	SECONDS	HOURS	MINUTES	SECONDS	
3.000000-05	1.000000-03	1.080000-01	3.000000-07	1.800000-05	1.080000-03	0
1.000000-04	6.000000-03	3.600000-01	1.000000-06	6.000000-05	3.600000-03	0
4.000000-04	2.400000-02	1.440000+00	3.000000-06	1.800000-04	1.080000-02	0
1.000000-03	6.000000-02	3.600000+00	1.000000-05	6.000000-04	3.600000-02	60
4.000000-03	2.400000-01	1.440000+01	2.000000-05	1.200000-03	7.200000-02	100
1.000000-02	6.000000-01	3.600000+01	3.000000-05	1.800000-03	1.080000-01	200
3.000000-02	1.800000+00	1.080000+02	4.000000-05	2.400000-03	1.440000-01	500

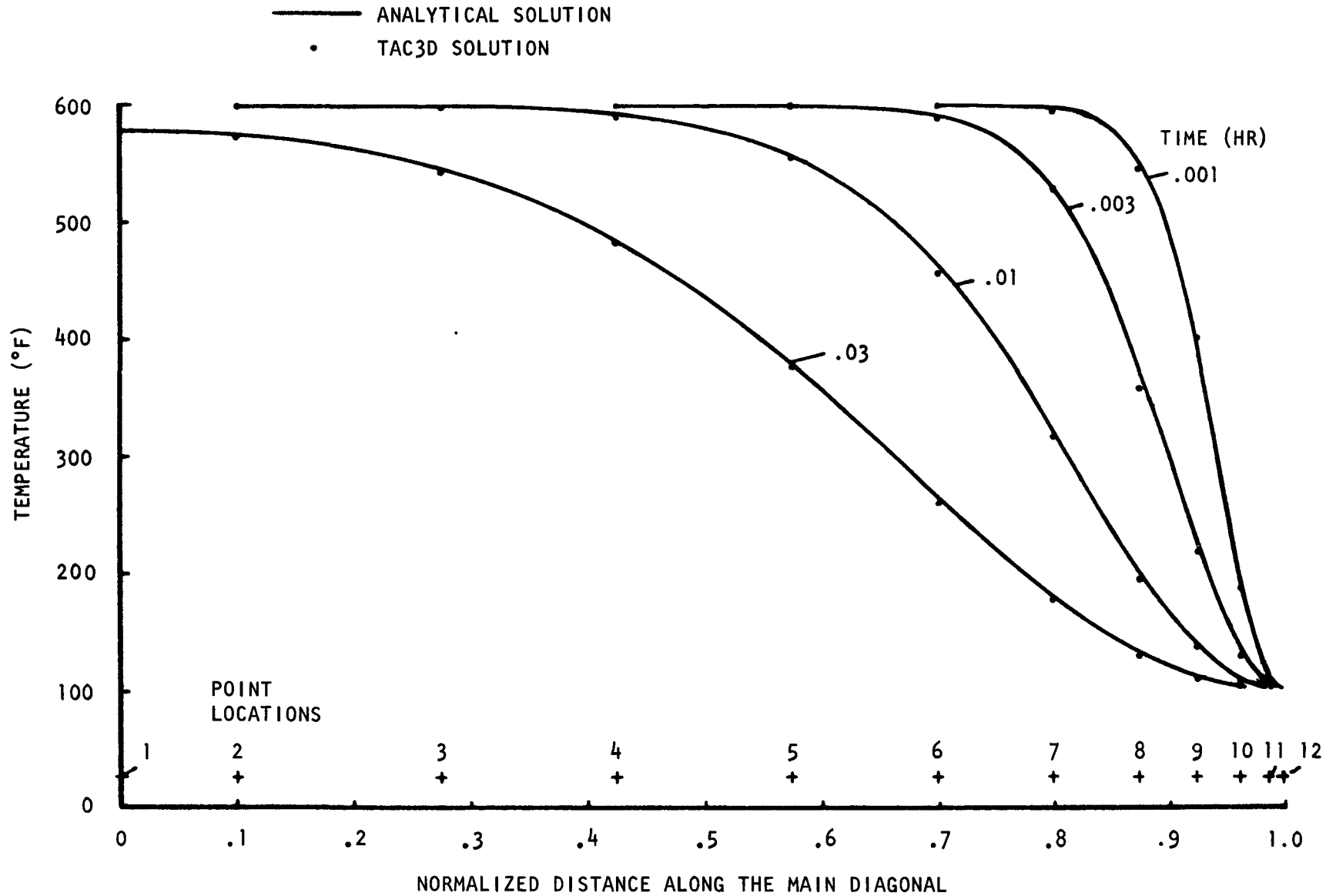


Fig. F.2. TAC3D sample problem — comparison of code solution with analytical solution

APPENDIX G

CODE ORGANIZATION AND FUNCTIONS OF SUBROUTINES IN TAC3D

The brief code description presented in Section 4 of this report is amplified in the outline below. The main program first calls Subroutine MP1 to read the input and calculate all constant data. It then calls Subroutine MP2, which in turn calls the computational subroutines continually until all iterations have been completed. All subroutines are called in the sequence listed below. When one subroutine is called by several other subroutines, its description is given at the point where it is first called.

The code contains several small subroutines which are internal to those described below. They are not mentioned since the functions they perform are contained within the descriptions of the subroutines of which they are a part.

MAIN - Main program of TAC3D

calls MP1
 MP2

A. Subroutine MP1 - Main program of Input Processing Section

called by MAIN
calls INPUT
 CHECK
 POINTS
 CONSTA
 GEOMET
 DPRINT
 INITEM
 FLOWCA
 TIME

EXTRA
PRETEM
OTCARD*

1. Subroutine INPUT - Reads and prints the title, geometry type, options and block description; reads and indexes the grid planes; indexes the blocks and block description data; prints a graphical display of the block boundaries.

called by MP1

calls ERROR, FCARD

- a. Subroutine ERROR - Prints an appropriate error message when input data is inconsistent or specifies a problem beyond code limitations.

called by all subroutines of input processing section
except CONSTA, EXTRA, DPRINT and OTCARD

calls None

- b. Subroutine FCARD - Maintains a record of input data for printing of approximate card images.

called by INPUT, INITEM, FLOWCA,
TIME, EXTRA, PRETEM

calls None

2. Subroutine CHECK - Checks for any inconsistencies between the grid planes and block boundaries. Identifies any material, coolant or gas numbers which are not allowed.

called by MP1

calls ERROR

3. Subroutine POINTS - Calculates and organizes geometrical data by calling BOUNDA; sets up an array of indicators identifying heat transfer configuration for each point adjacent to a gap plane.

called by MP1

calls BOUNDA, ERROR

*OTCARD is internal to MP1. It is mentioned individually because it performs a unique function not included in the above description of MP1.

- a. Subroutine BOUNDA - Calculates and indexes the point locations; identifies the highest and lowest point level indices for each direction in each block; identifies and indexes the gap planes; relates the gap plane indices to the grid plane indices.
called by POINTS
calls ERROR
4. Subroutine CONSTA - Calculates the effective path lengths between points; calculates the geometrical constants given in Appendix C, Tables C.1 through C.4.
called by MP1
calls None
5. Subroutine GEOMET - Calculates the node volumes and the geometrical components of the conductances between points as defined in Appendix B, Eqs. (B-5) through (B-12); prints all values of the above quantities if requested through input option.
called by MP1
calls ERROR, ARRAY1
 - a. Subroutine ARRAY1 - Prints a three-dimensional array.
called by GEOMET, PRETEM
calls None
6. Subroutine DPRINT - Prints the locations and indices of the points, grid planes and gap planes.
called by MP1
calls None
7. Subroutine INITEM - Reads specified initial temperature region description; assigns initial temperatures to all points according to these descriptions.
called by MP1
calls FCARD, ERROR
8. Subroutine FLOWCA - Reads and indexes the limits over which certain coolant thermal parameter functions apply. Prints these limits and the flow direction for each coolant.

called by MP1
calls FCARD, ERROR

9. Subroutine TIME - Reads, indexes and prints the data which specify the subdivision of the problem time scale and the frequency of printing results when the steady-state option is not being used.

called by MP1
calls FCARD, ERROR

10. Subroutine EXTRA - Reads and prints the function control constants described in Section 3.3.

called by MP1
calls FCARD

11. Subroutine PRETEM - Reads (if provided) the arrays of point and coolant terminal temperatures punched by the code from a preceding problem; replaces the specified initial temperatures read by INITEM with the above values; prints the initial temperatures to be used.

called by MP1
calls ERROR, FCARD, ARRAY1

12. Subroutine OTCARD - Prints approximate images of the input cards.

called by MP1
calls None

- B. Subroutine MP2 - Main program of Computational Section

called by MAIN
calls ERROR2
STEADY*
BLOCK
CONDUCT

*Called only when the steady-state option is being used.

STEP
COOL
SURT
PRINT
PUN

1. Subroutine ERROR2 - Terminates the problem and prints an appropriate error message when certain computational difficulties are encountered.

called by MP2, FLODAT, SURT

calls None
2. Subroutine STEADY - Calculates pseudo-capacitances and time steps and calls the temperature computation subroutines to obtain the steady-state solution as described in Appendix E.

called by MP2, PRINT
calls BLOCK, CONDOC, COOL, STEP,
SURT, PRINT, ARRAY, PUN
3. Subroutine BLOCK - Calls subroutines MADATA and FLODAT for each block on each iteration to obtain the current local values of the material and coolant thermal parameters.

called by MP2, STEADY
calls MADATA, FLODAT
 - a. Subroutine MADATA - Determines current local values of material and gas thermal parameters.

called by BLOCK, CONDOC
calls Functions which define material and gas thermal parameters as described in Section 3.2.

b. Subroutine FLODAT - Determines current local values of coolant thermal parameters.

called by BLOCK

calls BLKTYP and functions which define coolant thermal parameters as described in Section 3.2.

1) Subroutine BLKTYP - Determines those boundaries of a coolant block along which heat transfer coefficients are to be assigned.

called by FLODAT

calls None

4. Subroutine CONDOC - Calculates the composite conductivities by the methods developed in Appendix C; calls subroutine MADATA to determine the current local thermal conductivities of gases in gaps.

called by MP2, STEADY

calls MADATA

5. Subroutine STEP - Solves for the material point temperatures by the method outlined in Appendix A.

called by MP2, STEADY

calls None

6. Subroutine COOL - Solves for the coolant point and coolant terminal temperatures by the method developed in Appendix D.

called by MP2, STEADY

calls None

7. Subroutine SURT - Solves for the gap and coolant block surface temperatures by the methods developed in Appendix C.

called by MP2, STEADY

calls ERROR2

8. Subroutine PRINT - Prints the temperatures at all points for iterations specified in the input or for the smoothing iterations if the steady-state option is being used; prepares and prints all optional results which have been specified in the input.

called by MP2, STEADY

calls ARRAY, STEADY*, CUSTOM

- a. Subroutine ARRAY - Prints a three-dimensional array.

called by STEADY

calls None

- b. Subroutine CUSTOM - Performs any additional computations and printing according to Fortran V instructions supplied by the code user.

called by PRINT

calls As programmed by the code user

9. Subroutine PUN - Punches a card deck containing the arrays of point and coolant terminal temperatures ($^{\circ}$ R) resulting from the last iteration of the problem.

called by MP2, STEADY

calls None

*The call to STEADY is to obtain the residuals for the last smoothing iteration. These are always printed when the steady-state option is being used.



APPENDIX H

TAC3D CODE LISTING

The following pages are a card image listing of the TAC3D code. The elements are arranged in the calling sequence outlined in Appendix G. They are page indexed in alphabetical order below so that individual elements may be easily located.

<u>Element Name</u>	<u>Page</u>
ARRAY	237
ARRAY1	124
BLKTYP	187
BLOCK	154
BOUNDA	114
CDE (PDP procedure element)	78
CHECK	105
CONDOC	190
CONSTA	117
COOL	205
COOL1 through COOL15	188
CUSTOM	239
DPRINT	126
ERROR	103
ERROR2	147
EXTRA	139
FCARD	104
FLOWCA	132
FLODAT	171
FMAT1 through FMAT15	169
GEOMET	120
INITEM	128

<u>Element Name</u>	<u>Page</u>
INPUT	92
MADATA	157
MAIN	81
MP1	89
MP2	144
POINTS	107
PRETEM	140
PRINT	225
PUN	240
STEADY	148
STEP	202
SURT	215
TAC3D (processed map)	77
TIME	136

TAC3D (processed map)

```
      SEG  MAIN=*(MP1,MP2L)
MP2L  SEG  MP2-STEADY-PURJ-ARRAY=*(LNK1,LNK2,LNK3)
LNK1  SEG  COOL-STEP-CONDOC-BLOCK-SURT
LNK2  SEG  PRINT
LNK3  SEG  ERROR2
```


CDE (PDP procedure element)

```

COMDIM* FCOPY                                CDE 10
PARAMETER IQ=12,JQ=12,KQ=12,NQ=15,MQ=12,IGQ=4,JGQ=4,KGQ=4,LQ=50 CDE 20
===== CDE 30
C COMMON AND DIMENSION FOR RAT-3D MARCH, 1969 CDE 40
C CDE 50
REAL KR ,KZ ,KT CDE 60
LOGICAL FIRST ,GAS ,SW ,DP CDE 70
INTLGR OUTTAP CDE 80
C CDE 90
DIMENSION ,RBL ( LQ),RBH ( LQ),RDG ( LQ),MGR ( LQ), CDE 100
- ZBL ( LQ),ZBH ( LQ),ZDG ( LQ),MGZ ( LQ), CDE 110
- TBL ( LQ),TBH ( LQ),TDG ( LQ),MGT ( LQ), CDE 120
- NFLO ( NQ),CARD ( 1), CDE 130
- KR ( IQ,JQ,KQ), KZ ( IQ,JQ,KQ), CDE 140
- KT ( IQ,JQ,KQ), MT ( IQ,JQ,KQ) CDE 150
C CDE 160
COMMON /TEST / SW ( 25),OUTTAP ,DP ,NTA CDE 170
C CDE 180
COMMON /MXX / DTIME ( 21),FTIME ( 21),ITER ( 21),ITAPE ( 21) CDE 190
C CDE 200
COMMON /IXX / DELR ( IQ),RP ( IQ),RL ( IQ),RLN ( IQ), CDE 210
- IGR ( IQ) CDE 220
C CDE 230
COMMON /JXX / DELZ ( JQ),ZP ( JQ),ZL ( JQ),ZLN ( JQ), CDE 240
- JGZ ( JQ) CDE 250
C CDE 260
COMMON /KXX / DELT ( KQ),TP ( KQ),TL ( KQ),TLN ( KQ), CDE 270
- KGT ( KQ) CDE 280
C CDE 290
COMMON /NXX / FLOW ( NQ),TI ( NQ),TO ( NQ),IPATH ( NQ), CDE 300
- IFLO ( NQ),ITIN ( NQ),ITI ( NQ),ITO ( NQ), CDE 310
- IF ( NQ),FLODFP( NQ),TINDEP( NQ), CDE 320
- RLIM1 ( NQ),RLIM2 ( NQ),RLIM3 ( NQ),RLIM4 ( NQ), CDE 330
- FLIM1 ( NQ),FLIM2 ( NQ),FLIM3 ( NQ),FLIM4 ( NQ), CDE 340
- TLIM1 ( NQ),TLIM2 ( NQ),TLIM3 ( NQ),TLIM4 ( NQ), CDE 350
- RCPC ( NQ,MQ) CDE 360
C CDE 370
COMMON /IGXX / RATIOH(IGQ),RATIOK(IGQ),RATIOC(IGQ) CDE 380
C CDE 390
COMMON /JGXX / ZATIOH(JGQ),ZATIOK(JGQ) CDE 400
C CDE 410
COMMON /KGXX / TATIOH(KGQ),TATIOK(KGQ) CDE 420
C CDE 430
COMMON /LXX / IL ( LQ),IH ( LQ),JL ( LQ),JH ( LQ), CDE 440
- KL ( LQ),KH ( LQ),MB ( LQ) CDE 450

```

C	COMMON /USER /	DT	,DR	,FTR	,FTZ	,CDE	460
-		FTT	,GK	,HR	,FR	,CDE	470
-		RE	,ST	,TOUT	,TIN	,CDE	480
C	COMMON /XMAX /	MAXFLO	,MMAX	,MAXRP	,MAXZP	,CDE	490
-		MAXTP	,MAXNB	,MAXMAT	,MAXRG	,CDE	500
-		MAXZG	,MAXTG			,CDE	510
C	COMMON /XSUBS /	IM	,IM1	,IMAX	,IGLS	,CDE	520
-		IGHS	,NRG	,LMAX	,JM	,CDE	530
-		JM1	,JMAX	,JGLS	,JGHS	,CDE	540
-		NZG	,KM	,KM1	,KMAX	,CDE	550
-		KGLS	,KGHS	,NTG		,CDE	560
C	COMMON /MISCXX/	CURTI	,FIRST	,GAS	,ICOUNT	,CDE	570
-		ISHAPE	,NITER	,NPRINT	,SCALE	,CDE	580
-		NI	,TB	,TH	,DATI	,CDE	590
-		NC	,IERROR(2)	,ZA (14)	,PNAME (12)	,CDE	600
-		CS1	,CS2	,CS3		,CDE	610
C	COMMON /XMADAX/	ILS	,IHS	,JLS	,JHS	,CDE	620
-		KLS	,KHS			,CDE	630
C	COMMON /XTRA /	A1	,A2	,A3	,A4	,CDE	640
-		A5	,A6	,A7	,A8	,CDE	650
-		A9	,A10	,A11	,A12	,CDE	660
-		A13	,A14	,A15	,A16	,CDE	670
-		A17	,A18			,CDE	680
C	COMMON /IGXJXK/	GAPR (IGQ,JQ,KQ),		MATRG (IGQ,JQ,KQ),		,CDE	690
-		RBBTL (IGQ,JQ,KQ),		RBBTH (IGQ,JQ,KQ),		,CDE	700
-		REML (IGQ,JQ,KQ),		REMH (IGQ,JQ,KQ),		,CDE	710
-		RATIOB(IGQ,JQ,KQ)				,CDE	720
C	COMMON /JGXIXK/	GAPZ (JGQ,IQ,KQ),		MATZG (JGQ,IQ,KQ),		,CDE	730
-		ZBBTL (JGQ,IQ,KQ),		ZBBTH (JGQ,IQ,KQ),		,CDE	740
-		ZEML (JGQ,IQ,KQ),		ZEMH (JGQ,IQ,KQ),		,CDE	750
-		ZATIOB(JGQ,IQ,KQ)				,CDE	760
C	COMMON /KGXIXJ/	GAPT (KGQ,IQ,JQ),		MATTG (KGQ,IQ,JQ),		,CDE	770
-		TBBTL (KGQ,IQ,JQ),		TBBTH (KGQ,IQ,JQ),		,CDE	780
-		TEML (KGQ,IQ,JQ),		TEMH (KGQ,IQ,JQ),		,CDE	790
-		TATIOB(KGQ,IQ,JQ)				,CDE	800
C	COMMON /IXJXX/	RR (IQ,JQ,KQ),		RZ (IQ,JQ,KQ),		,CDE	810
						,CDE	820
						,CDE	830
						,CDE	840
						,CDE	850
						,CDE	860
						,CDE	870
						,CDE	880
						,CDE	890
						,CDE	900
						,CDE	910

```

-          RT   ( IQ,JQ,KQ),      V   ( IQ,JQ,KQ),      CDE  920
-          W   ( IQ,JQ,KQ),      CONR ( IQ,JQ,KQ),      CDE  930
-          CONZ ( IQ,JQ,KQ),      CONT ( IQ,JQ,KQ),      CDE  940
-          RCP  ( IQ,JQ,KQ),      T   ( IQ,JQ,KQ),      CDE  950
-          TT   ( IQ,JQ,KQ)                CDE  960
C          EQUIVALENCE (MT   ,W   ),(RBBTL ,RBL   ),(RBBTH ,RBH   ), CDE  9A0
-          (REML ,RDG  ),(REMH  ,MGR  ),(ZBBTL ,ZBL  ), CDE  990
-          (ZBBTH ,ZBH  ),(ZEML  ,ZDG  ),(ZEMH  ,MGZ  ), CDE 1000
-          (TBBTL ,TBL  ),(TBBTH ,TBH  ),(TEML  ,TDG  ), CDE 1010
-          (TEMH  ,MGT  ),(CONR  ,KR   ),(CONZ  ,KZ   ), CDE 1020
-          (CONT  ,KT   ),(CONR  ,CARD ),(FLODEP ,NFLO ) CDE 1030
C=====CDE 1040
C=====CDE 1050
END                                           CDE 1060

```

```

=====MAIN 10
C      INCLUDE      COMDIM      MAIN 20
=====MAIN 30
C      MAIN PROGRAM OF TAC3D      MAIN 40
=====MAIN 50
C      DEFINITIONS OF PROGRAM VARIABLES      MAIN 70
=====MAIN 80
C      A1...A18      EXTRA FUNCTION VARIABLES USED IN MATERIAL      MAIN 90
C      AND COOLANT FUNCTIONS      MAIN 100
=====MAIN 110
C      CONK(I,J,K)      INVERSE LOCAL RADIAL THERMAL CONDUCTIVITY      MAIN 120
C      (HR-FT-R/BTU)
C      CONT(I,J,K)      INVERSE LOCAL THETA THERMAL CONDUCTIVITY      MAIN 130
C      (HR-FT-R/BTU)
C      CONZ(I,J,K)      INVERSE LOCAL AXIAL THERMAL CONDUCTIVITY      MAIN 160
C      (HR-FT-R/BTU)
C      CURTI      CURRENT TIME, (HR)      MAIN 170
C      MAIN 180
=====MAIN 190
C      DATI      TIME AT WHICH THE TIMEDEPENDENT DATA SHOULD      MAIN 200
C      BE CALCULATED      MAIN 210
C      DT      CURRENT TIME INCREMENT, (HR)      MAIN 220
C      DTIME(M)      TIME STEP USED IN TIME PERIOD M, HR.      MAIN 230
C      DIM(2)      CONSTANT DIMENSION VALUES IN HEAT TRANSFER      MAIN 240
C      COEFFICIENT DETERMINATION      MAIN 250
C      OR      CONTAINS THE LOCAL TEMPERATURE IN DEGREES      MAIN 260
C      RANKINE. IT IS USED IN THE DATA FUNCTIONS.      MAIN 270
C      DELR(I)      DISTANCE BETWEEN POINTS I AND I+1(FT)      MAIN 280
C      DELZ(J)      DISTANCE BETWEEN POINTS J AND J+1(FT)      MAIN 290
C      DELT(K)      DISTANCE BETWEEN POINTS K AND K+1(FT)      MAIN 300
=====MAIN 310
C      FTIME(M)      TIME AT WHICH TIME PERIOD M ENDS, HR.      MAIN 320
C      FR = FLOW(N)      LOCAL FLOW RATE, LB/HR, USED IN DATA FUNCTIONS      MAIN 330
C      FIRSI      A LOGICAL PARAMETER CONTROLLING CERTAIN ACTIONS      MAIN 340
C      TRUE= SPECIAL ACTION, ONLY ONCE      MAIN 350
C      FALSE= NORMAL ACTION, ALL OTHER TIMES      MAIN 360
C      FLOW(N)      FLOW RATE OF COOLANT N, LB/HR      MAIN 370
C      =U, CURRENT COOLANT NOT USED      MAIN 380
C      FLIM1(N)      LOW LIMIT, LOW RANGE FLOW RATE FUNCTION      MAIN 390
C      FLIM2(N)      UPPER LIMIT, LOW RANGE FLOW RATE FUNCTION      MAIN 400
C      FLIM3(N)      UPPER LIMIT, MIDDLE RANGE FLOW RATE FUNCTION      MAIN 410
C      FLIM4(N)      UPPER LIMIT, UPPER RANGE FLOW RATE FUNCTION      MAIN 420
C      FTR = RP(I)      RADIAL LOCATION OF CURRENT POINT, FT      MAIN 430
C      FTT = TP(K)      THETA LOCATION OF CURRENT POINT, FT      MAIN 440
C      FTZ = ZP(J)      AXIAL LOCATION OF THE CURRENT POINT, FT      MAIN 450

```

```

C=====MAIN 460
C   GAPR(IG,J,K)   RADIAL GAP INDICATOR           MAIN 470
C                 =POSITIVE, LOCAL RADIAL GAP WIDTH IN GAPLINE IG   MAIN 480
C                 =0.0, NO GAP OR COOLANT           MAIN 490
C                 =-1.0E-10, COOLANT ON HIGH INDEX SIDE OF         MAIN 500
C                 GAPLINE IG                               MAIN 510
C                 =-2.0E-10, COOLANT ON LOW INDEX SIDE OF         MAIN 520
C                 GAPLINE IG                               MAIN 530
C                 =-3.0E-10, TWO COOLANTS ADJACENT AT GAPLINE IG   MAIN 540
C   GAPZ(JG,I,K)   AXIAL GAP INDICATOR           MAIN 550
C                 =POSITIVE, LOCAL AXIAL GAP WIDTH IN GAPLINE JG   MAIN 560
C                 =0.0, NO GAP OR COOLANT           MAIN 570
C                 =-1.0E-10, COOLANT ON HIGH INDEX SIDE OF         MAIN 580
C                 GAPLINE JG                               MAIN 590
C                 =-2.0E-10, COOLANT ON LOW INDEX SIDE OF         MAIN 600
C                 GAPLINE JG                               MAIN 610
C                 =-3.0E-10, TWO COOLANTS ADJACENT AT GAPLINE JG   MAIN 620
C   GAPL(KG,I,J)   THETA GAP INDICATOR           MAIN 630
C                 =POSITIVE, LOCAL THETA GAP WIDTH IN GAPLINE KG   MAIN 640
C                 =0.0, NO GAP OR COOLANT           MAIN 650
C                 =-1.0E-10, COOLANT ON HIGH INDEX SIDE OF         MAIN 660
C                 GAPLINE KG                               MAIN 670
C                 =-2.0E-10, COOLANT ON LOW INDEX SIDE OF         MAIN 680
C                 GAPLINE KG                               MAIN 690
C                 =-3.0E-10, TWO COOLANTS ADJACENT AT GAPLINE KG   MAIN 700
C   GAS           LOGICAL VARIABLE , TRUE IF GAS           MAIN 710
C   GK           LOCAL GAS CONDUCTIVITY, (BTU/HR,FT,F)       MAIN 720
C=====MAIN 730
C   HR           CONTAINS THE TIME IN HOURS. IT IS USED IN     MAIN 740
C                 THE DATA FUNCTIONS.                   MAIN 750
C=====MAIN 760
C   IGR(I)       GAP INDEX OF GAP AT LINE I                 MAIN 770
C   IM1          IM-1                                       MAIN 780
C   I            RADIAL POINT OR LINE INDEX                 MAIN 790
C   IA          LOCAL HEAT TRANSFER CORRELATION RANGE       MAIN 800
C   ITER(M)     NUMBER OF TIME STEPS BETWEEN DATA EVALUATIONS MAIN 810
C                 IN TIME PERIOD M                         MAIN 820
C   IUNIT       CODE NUMBER INDICATING THE UNITS OF THE CURRENT MAIN 830
C                 TIME PERIOD DATA BEING READ.          MAIN 840
C                 0=SAME UNITS AS BEFORE                 MAIN 850
C                 1=SEC.                                  MAIN 860
C                 2=MIN.                                  MAIN 870
C                 3=HR.                                   MAIN 880
C   IFLO(N)     FLOW DEPENDENCE NUMBER OF COOLANT N        MAIN 890
C                 =1, NO DEPENDENCE                     MAIN 900
C                 =2, FLOW DEPENDS ON TIME               MAIN 910

```

C		=3, FLOW DEPENDS ON COOLANT OUTLET TEMPERATURE	MAIN 920
C		=4, FLOW DEPENDS ON COOLANT INLET TEMPERATURE	MAIN 930
C	IG	RADIAL GAP INDEX	MAIN 940
C	IGLS	THE GAP NUMBER ASSOCIATED WITH THE LOW RADIAL	MAIN 950
C		BOUNDARY OF THE CURRENT BLOCK	MAIN 960
C	IGHS	THE GAP NUMBER ASSOCIATED WITH THE HIGH RADIAL	MAIN 970
C		BOUNDARY OF THE CURRENT BLOCK	MAIN 980
C	IERROR(2)	NAME OF ROUTINE IN WHICH THE ERROR WAS CAUGHT	MAIN 990
C	ICOUNT	NUMBER OF RECORDS ON THE EXTRA OUTPUT TAPE	MAIN,000
C	ITAPE(M)	NUMBER OF Timesteps BETWEEN TEMPERATURE	MAIN1010
C		PRINTS IN PERIOD M	MAIN1020
C	IL(L)	INDEX OF THE LOW RADIAL POINT OF BLOCK L	MAIN1030
C	IH(L)	INDEX OF THE HIGH RADIAL POINT OF BLOCK L	MAIN1040
C	IHS = IH(L)	INDEX OF THE HIGH RADIAL POINT OF CURRENT BLOCK	MAIN1050
C	ILS = IL(L)	INDEX OF THE LOW RADIAL POINT OF CURRENT BLOCK	MAIN1060
C	IM	NUMBER OF RADIAL GRIDLINES	MAIN1070
C	IMAX	NUMBER OF RADIAL POINTS	MAIN1090
C	ISHAPE	=1 FOR X-Y-Z GEOMETRY	MAIN1090
C		=0 FOR R-T-Z GEOMETRY	MAIN1100
C	ITIN(N)	COOLANT INLET TEMPERATURE DEPENDENCE NUMBER	MAIN1110
C		=1, NO DEPENDENCE	MAIN1120
C		=2, COOLANT INLET TEMPERATURE DEPENDS ON TIME	MAIN1130
C		=3, COOLANT INLET TEMPERATURE DEPENDS ON FLOW	MAIN1140
C		RATE	MAIN1150
C		=4, COOLANT INLET TEMPERATURE DEPENDS ON	MAIN1160
C		OUTLET TEMPERATURE	MAIN1170
C	IT = ITIN(N)	LOCAL FLOW RATE DEPENDENCE	MAIN1180
C	IF(N)	INTEGER VALUE OF FLOW(N)	MAIN1190
C	ITI(N)	INTEGER VALUE OF TI(N)	MAIN1200
C	ITO(N)	INTEGER VALUE OF TO(N)	MAIN1210
C	IPATH(N)	FLOW DIRECTION FOR COOLANT	MAIN1220
C		=1, RADIAL DIRECTION	MAIN1230
C		=2, AXIAL DIRECTION	MAIN1240
C		=3, THETA DIRECTION	MAIN1250
C		+ = FLOW IN DIRECTION OF INCREASING INDEX	MAIN1260
C		- = FLOW IN DIRECTION OF DECREASING INDEX	MAIN1270
C		=====	MAIN1280
C	J	AXIAL POINT OR LINE INDEX	MAIN1290
C	JG	AXIAL GAP INDEX	MAIN1300
C	JM1	JM-1	MAIN1310
C	JGZ(J)	GAP INDEX OF GAP AT LINE J	MAIN1320
C	JGLS	THE GAP NUMBER ASSOCIATED WITH THE LOW AXIAL	MAIN1330
C		BOUNDARY OF THE CURRENT BLOCK	MAIN1340
C	JGHS	THE GAP NUMBER ASSOCIATED WITH THE HIGH AXIAL	MAIN1350
C		BOUNDARY OF THE CURRENT BLOCK	MAIN1360
C	JL(L)	INDEX OF THE LOW AXIAL POINT OF BLOCK L	MAIN1370

C	JH(L)	INDEX OF THE HIGH AXIAL POINT OF BLOCK L	MAIN1380
C	JHS = JH(L)	INDEX OF THE HIGH AXIAL POINT OF CURRENT BLOCK	MAIN1390
C	JLS = JL(L)	INDEX OF THE LOW AXIAL POINT OF CURRENT BLOCK	MAIN1400
C	JM	NUMBER OF AXIAL GRIDLINES	MAIN1410
C	JMAX	NUMBER OF AXIAL POINTS	MAIN1420
=====			
C	K	THETA POINT OR LINE INDEX	MAIN1430
C	KGT(K)	GAP INDEX OF GAP AT LINE K	MAIN1440
C	KR(I,J,K)	AVERAGE RADIAL CONDUCTIVITY BETWEEN POINTS I AND	MAIN1450
C		I+1	MAIN1460
C		(BTU/(HR,FT,F))	MAIN1470
C	KZ(I,J,K)	AVERAGE AXIAL CONDUCTIVITY BETWEEN POINTS J AND	MAIN1480
C		J+1	MAIN1490
C		(BTU/(HR,FT,F))	MAIN1500
C	KT(I,J,K)	AVERAGE THETA CONDUCTIVITY BETWEEN POINTS K AND	MAIN1510
C		K+1	MAIN1520
C		(BTU/(HR,FT,F))	MAIN1530
C	KM1	KM-1	MAIN1540
C	KG	THETA GAP INDEX	MAIN1550
C	KGLS	THE GAP NUMBER ASSOCIATED WITH THE LOW THETA	MAIN1560
C		BOUNDARY OF THE CURRENT BLOCK	MAIN1570
C	KGHS	THE GAP NUMBER ASSOCIATED WITH THE HIGH THETA	MAIN1580
C		BOUNDARY OF THE CURRENT BLOCK	MAIN1590
C	KH(L)	INDEX OF THE HIGH THETA POINT OF BLOCK L	MAIN1600
C	KL(L)	INDEX OF THE LOW THETA POINT OF BLOCK L	MAIN1610
C	KHS = KH(L)	INDEX OF THE HIGH THETA POINT OF CURRENT BLOCK	MAIN1620
C	KLS = KL(L)	INDEX OF THE LOW THETA POINT OF CURRENT BLOCK	MAIN1630
C	KM	NUMBER OF THETA GRIDLINES	MAIN1640
C	KMAX	NUMBER OF THETA POINTS	MAIN1650
=====			
C	LMAX	NUMBER OF BLOCKS	MAIN1660
=====			
C	MT(I,J,K)	TABLE USED TO TEST COMPLETENESS OF THE	MAIN1670
C		BLOCK ASSIGNMENTS	MAIN1680
C	MAXFLO	MAXIMUM NUMBER OF COOLANTS ALLOWED	MAIN1690
C	MMAX	MAXIMUM NUMBER OF TIME PERIODS ALLOWED	MAIN1700
C	MAXRP	MAXIMUM NUMBER OF RADIAL POINTS ALLOWED	MAIN1710
C	MAXZP	MAXIMUM NUMBER OF AXIAL POINTS ALLOWED	MAIN1720
C	MAXIP	MAXIMUM NUMBER OF THETA POINTS ALLOWED	MAIN1730
C	MAXNB	MAXIMUM NUMBER OF BLOCKS ALLOWED	MAIN1740
C	MGR(L)	MATERIAL NUMBER OF THE RADIAL GAP OF BLOCK L	MAIN1750
C	MGZ(L)	MATERIAL NUMBER OF THE AXIAL GAP OF BLOCK L	MAIN1760
C	MGT(L)	MATERIAL NUMBER OF THE THETA GAP OF BLOCK L	MAIN1770
C	MAXMAT	MAXIMUM NUMBER OF MATERIALS AND COOLANTS ALLOWED	MAIN1780
C	MAXRG	MAXIMUM NUMBER OF RADIAL GAPS	MAIN1790
C	MAXZG	MAXIMUM NUMBER OF AXIAL GAPS	MAIN1800

C	MAXIG	MAXIMUM NUMBER OF THETA GAPS	MAIN1840
C	M = IPATH(N)	LOCAL FLOW DIRECTION	MAIN1850
C	MATKG(I,G,J,K)	1. NUMBER OF GAS IN GAP ALONG RADIAL GAPLINE IG	MAIN1860
C		AT AXIAL LEVEL J, THETA LEVEL K OR,	MAIN1870
C		2. INDICATOR DENOTING STATUS OF RADIATION ON	MAIN1880
C		RADIAL COOLANT BOUNDARY (GAPLINE) IG AT AXIAL	MAIN1890
C		LEVEL J, THETA LEVEL K	MAIN1900
C		=100, NO RADIATION SPECIFIED	MAIN1910
C		=200, RADIATION SPECIFIED AND INCLUDED	MAIN1920
C		=300, RADIATION SPECIFIED BUT TEMPORARILY	MAIN1930
C		EXCLUDED IN ORDER TO COMPLETE AN ITERATION	MAIN1940
C	MATZG(J,G,I,K)	1. NUMBER OF GAS IN GAP ALONG AXIAL GAPLINE JG	MAIN1950
C		AT RADIAL LEVEL I, THETA LEVEL K OR,	MAIN1960
C		2. INDICATOR DENOTING STATUS OF RADIATION ON	MAIN1970
C		AXIAL COOLANT BOUNDARY (GAPLINE) JG AT RADIAL	MAIN1980
C		LEVEL I, THETA LEVEL K	MAIN1990
C		=100, NO RADIATION SPECIFIED	MAIN2000
C		=200, RADIATION SPECIFIED AND INCLUDED	MAIN2010
C		=300, RADIATION SPECIFIED BUT TEMPORARILY	MAIN2020
C		EXCLUDED IN ORDER TO COMPLETE AN ITERATION	MAIN2030
C	MATTG(K,G,I,J)	1. NUMBER OF GAS IN GAP ALONG THETA GAPLINE KG	MAIN2040
C		AT RADIAL LEVEL I, AXIAL LEVEL J OR,	MAIN2050
C		2. INDICATOR DENOTING STATUS OF RADIATION ON	MAIN2060
C		THETA COOLANT BOUNDARY (GAPLINE) KG AT RADIAL	MAIN2070
C		LEVEL I, AXIAL LEVEL J	MAIN2080
C		=100, NO RADIATION SPECIFIED	MAIN2090
C		=200, RADIATION SPECIFIED AND INCLUDED	MAIN2100
C		=300, RADIATION SPECIFIED BUT TEMPORARILY	MAIN2110
C		EXCLUDED IN ORDER TO COMPLETE AN ITERATION	MAIN2120
C	MB(L)	MATERIAL NUMBER OF BLOCK L	MAIN2130
C		IF COOLANT, COOLANT NUMBER STORED AS NEGATIVE	MAIN2140
C		MATERIAL NUMBER	MAIN2150
C	=====		
C	N	COOLANT SUBSCRIPT	MAIN2160
C	NRG	NUMBER OF RADIAL GAPS	MAIN2180
C	NZG	NUMBER OF AXIAL GAPS	MAIN2190
C	NTG	NUMBER OF THETA GAPS	MAIN2200
C	NI	MATERIAL SUBSCRIPT	MAIN2210
C	NITER	THE CURRENT ITERATION NUMBER, ONE RADIAL, AXIAL,	MAIN2220
C		AND THETA PASS ARE COUNTED AS ONE ITERATION.	MAIN2230
C	NPRINT	PRINTOUT DIRECTION DEPENDENCE	MAIN2240
C		=1, PRINT IN THETA PLANES	MAIN2250
C		=2, PRINT IN AXIAL PLANES	MAIN2260
C		=3, PRINT IN RADIAL PLANES	MAIN2270
C	NUMBER(25)	LIST OF INTEGERS FROM 1 TO 25 USED IN PRINTOUT	MAIN2280
C	=====		
C			MAIN2290

C	RDG(L)	RADIAL GAP THICKNESS (FT)	MAIN2300
C	RBL(L)	LOW RADIAL BOUNDARY OF BLOCK L (FT)	MAIN2310
C	RBH(L)	HIGH RADIAL BOUNDARY OF BLOCK L (FT)	MAIN2320
C	RL(I)	RADIAL OR X-GRIDLINE(FT)	MAIN2330
C	RATIOB(IG,J,K)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN2340
C	RLN(I)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN2360
C	RATIOH(IG)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN2370
C	RATIOK(IG)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN2390
C	RATIOK(IG)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN2400
C	RBBIH(IG,J,K)	LOCAL TEMPERATURE OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH I INDEX SIDE BY GAP IG,(R)	MAIN2420
C	RBBTL(IG,J,K)	LOCAL TEMPERATURE OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG,(R)	MAIN2430
C	RBBTL(IG,J,K)	LOCAL TEMPERATURE OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG,(R)	MAIN2440
C	RBBTL(IG,J,K)	LOCAL TEMPERATURE OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG,(R)	MAIN2450
C	RBBTL(IG,J,K)	LOCAL TEMPERATURE OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG,(R)	MAIN2460
C	RBBTL(IG,J,K)	LOCAL TEMPERATURE OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG,(R)	MAIN2470
C	RCP(I,J,K)	SPECIFIC HEAT OF VOLUME I,J,K(BTU/F)	MAIN2480
C	RCPC(N,I)	SPECIFIC HEAT COOLANT N, LEVEL I IN FLOW DIRECTION	MAIN2490
C	RE	LOCAL REYNOLDS NUMBER, USED IN DATA FUNCTIONS	MAIN2500
C	REMH(IG,J,K)	INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH I INDEX SIDE BY GAP IG	MAIN2510
C	REMH(IG,J,K)	INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH I INDEX SIDE BY GAP IG	MAIN2520
C	REML(IG,J,K)	INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG	MAIN2530
C	REML(IG,J,K)	INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG	MAIN2540
C	REML(IG,J,K)	INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG	MAIN2550
C	REML(IG,J,K)	INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX SIDE BY GAP IG	MAIN2560
C	RLIM1(N)	LOWER LIMIT, LOW RANGE HEAT TRANSFER CORRELATION	MAIN2570
C	RLIM2(N)	UPPER LIMIT, LOW RANGE HEAT TRANSFER CORRELATION	MAIN2580
C	RLIM3(N)	UPPER LIMIT, MIDDLE RANGE HEAT TRANSFER CORRELATION	MAIN2590
C	RLIM3(N)	UPPER LIMIT, MIDDLE RANGE HEAT TRANSFER CORRELATION	MAIN2600
C	RLIM4(N)	UPPER LIMIT, UPPER RANGE HEAT TRANSFER CORRELATION	MAIN2610
C	RLIM4(N)	UPPER LIMIT, UPPER RANGE HEAT TRANSFER CORRELATION	MAIN2620
C	RLIM4(N)	UPPER LIMIT, UPPER RANGE HEAT TRANSFER CORRELATION	MAIN2630
C	RP(I)	RADIAL LOCATION OF POINT I,(FT)	MAIN2630
C	RR(I,J,K)	INVERSE RADIAL RESISTANCE BETWEEN POINTS I AND I+1(FT)	MAIN2640
C	RR(I,J,K)	INVERSE RADIAL RESISTANCE BETWEEN POINTS I AND I+1(FT)	MAIN2650
C	RZ(I,J,K)	INVERSE AXIAL RESISTANCE BETWEEN POINTS J AND J+1(FT)	MAIN2660
C	RZ(I,J,K)	INVERSE AXIAL RESISTANCE BETWEEN POINTS J AND J+1(FT)	MAIN2670
C	RT(I,J,K)	INVERSE THETA RESISTANCE BETWEEN POINTS K AND K+1(FT)	MAIN2680
C	RT(I,J,K)	INVERSE THETA RESISTANCE BETWEEN POINTS K AND K+1(FT)	MAIN2690
C	ST	LOCAL SURFACE TEMPERATURE, R, USED IN DATA FUNCTIONS	MAIN2700
C	SCALE	CONVERSION FACTOR	MAIN2720
C		=12.0 FOR X-Y-Z GEOMETRY	MAIN2730
C		= 57.2958 FOR R-Z-T GEOMETRY	MAIN2740
C			MAIN2750

C		=57.296 FOR R-T-Z GEOMETRY	MAIN2760
C	TL(K)	THETA OR Y-GRIDLINE(FT)	MAIN2770
C	TBH(L)	HIGH THETA BOUNDARY OF BLOCK L (FT OR RAD)	MAIN2780
C	TBL(L)	LOW THETA BOUNDARY OF BLOCK L (FT OR RAD)	MAIN2790
C	TDG(L)	THETA GAP THICKNESS (FT OR RAD)	MAIN2800
C	T(I,J,K)	LOCAL TEMPERATURE OF POINT I,J,K	MAIN2810
C	TI(I,J,K)	INTEGER VALUE OF T(I,J,K)	MAIN2820
C	TBBIH(KG,I,J)	LOCAL TEMPERATURE OF THE THETA BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH K INDEX SIDE	MAIN2830
C		BY GAP KG,(R)	MAIN2840
C	TBBIL(KG,I,J)	LOCAL TEMPERATURE OF THE THETA BOUNDARY OF A BLOCK BOUNDED ON ITS LOW K INDEX SIDE	MAIN2850
C		BY GAP KG,(R)	MAIN2860
C	TEMH(KG,I,J)	INVERSE OF THE LOCAL EMISSIVITY OF THE THETA BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH K INDEX	MAIN2880
C		SIDE BY GAP KG	MAIN2890
C	TEML(KG,I,J)	INVERSE OF THE LOCAL EMISSIVITY OF THE THETA BOUNDARY OF A BLOCK BOUNDED ON ITS LOW K INDEX	MAIN2900
C		SIDE BY GAP KG	MAIN2910
C	TLIM1(N)	LOW LIMIT, LOW RANGE INLET TEMPERATURE FUNCTION	MAIN2940
C	TLIM2(N)	HIGH LIMIT, LOW RANGE INLET TEMPERATURE FUNCTION	MAIN2950
C	TLIM3(N)	HIGH LIMIT, MIDDLE RANGE INLET TEMPERATURE FUNCTION	MAIN2960
C			MAIN2970
C	TLIM4(N)	HIGH LIMIT, UPPER RANGE INLET TEMPERATURE FUNCTION	MAIN2980
C			MAIN2990
C	TOU1 = TO(N)	LOCAL COOLANT OUTLET TEMPERATURE, R, USED IN DATA FUNCTIONS.	MAIN3000
C			MAIN3010
C	TIN = TI(N)	LOCAL COOLANT INLET TEMPERATURE, R, USED IN DATA FUNCTIONS	MAIN3020
C			MAIN3030
C	TO(N)	COOLANT OUTLET TEMPERATURE	MAIN3040
C	TI(N)	COOLANT INLET TEMPERATURE	MAIN3050
C	TH	LOCAL WALL TEMPERATURE,(R).	MAIN3060
C	TL	LOCAL WALL TEMPERATURE,(R).	MAIN3070
C	TP(K)	THETA LOCATION OF POINT K, (FT)	MAIN3080
C	TATIOK(KG)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN3090
C			MAIN3100
C	TATIOH(KG)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN3110
C			MAIN3120
C	TLN(K)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN3130
C			MAIN3140
C	TATIOB(KG,I,J)	GEOMETRICAL CONSTANT USED TO CALCULATE THE EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MAIN3150
C			MAIN3160
C			MAIN3170
C	=====	=====	MAIN3180
C	w(I,J,K)	HEAT GENERATION OF VOLUME I,J,K(BTU/HR)	MAIN3190
C	=====	=====	MAIN3200
C	V(I,J,K)	VOLUME ASSOCIATED WITH POINT I,J,K	MAIN3210

```

=====MAIN3200
C      X1          CURRENT TIME (MINUTES)          MAIN3210
C      X2          CURRENT TIME (SECONDS)         MAIN3240
=====MAIN3250
C      ZBL(L)     LOW AXIAL BOUNDARY OF BLOCK L (FT) MAIN3260
C      ZL(J)      AXIAL GRIDLINE (FT)            MAIN3270
C      ZBH(L)     HIGH AXIAL BOUNDARY OF BLOCK L (FT) MAIN3280
C      ZDG(L)     AXIAL GAP THICKNESS (FT)       MAIN3290
C      ZBBTH(JG,I,K) LOCAL TEMPERATURE OF THE AXIAL BOUNDARY OF A MAIN3300
C                      BLOCK BOUNDED ON ITS HIGH J INDEX SIDE MAIN3310
C                      BY GAP JG,(R)             MAIN3320
C      ZBBTL(JG,I,K) LOCAL TEMPERATURE OF THE AXIAL BOUNDARY OF A MAIN3330
C                      BLOCK BOUNDED ON ITS LOW J INDEX SIDE MAIN3340
C                      BY GAP JG,(R)             MAIN3350
C      ZEMH(JG,I,K) INVERSE OF THE LOCAL EMISSIVITY OF THE AXIAL MAIN3360
C                      BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH J INDEX MAIN3370
C                      SIDE BY GAP JG           MAIN3380
C      ZEML(JG,I,K) INVERSE OF THE LOCAL EMISSIVITY OF THE AXIAL MAIN3390
C                      BOUNDARY OF A BLOCK BOUNDED ON ITS LOW J INDEX MAIN3400
C                      SIDE BY GAP JG           MAIN3410
C      ZP(J)      AXIAL LOCATION OF POINT J,(FT) MAIN3420
C      ZAT10K(JG) GEOMETRICAL CONSTANT USED TO CALCULATE THE MAIN3430
C                      EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3440
C      ZAT10H(JG) GEOMETRICAL CONSTANT USED TO CALCULATE THE MAIN3450
C                      EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3460
C      ZAT10B(JG,I,K) GEOMETRICAL CONSTANT USED TO CALCULATE THE MAIN3470
C                      EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3480
C      ZLN(J)     GEOMETRICAL CONSTANT USED TO CALCULATE THE MAIN3490
C                      EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3500
=====MAIN3510
=====MAIN3520
=====MAIN3530
C      READ AND PROCESS THE INPUT DATA OF THE NEXT PROBLEM MAIN3540
C      20 CALL MP1 MAIN3550
C      HAS ANY ERROR BEEN FOUND? -YES,NO-- MAIN3560
C      IF (IERROR(1).NE.0) GO TO 20 MAIN3570
C      CALCULATE AND PRINT THE TEMPERATURE DISTRIBUTIONS MAIN3580
C      CALL MP2 MAIN3590
C      GO TO 20 MAIN3600
=====MAIN3610
C      END MAIN3620

```

SUBROUTINE MPI	MP1	10
INCLUDE COMDIM	MP1	20
C=====	MP1	30
CB READ AND PROCESS THE INPUT DATA OF ONE PROBLEM	MP1	40
C=====	MP1	50
C=====	MP1	60
C ASSIGN THE FIXED CONSTANTS	MP1	70
OUTTAP=15	MP1	80
MAXKP=IQ	MP1	90
MAXZP=JQ	MP1	100
MAXTP=KQ	MP1	110
MAXKG=IQQ	MP1	120
MAXZG=JQQ	MP1	130
MAXTG=KQQ	MP1	140
MAXMAT=15	MP1	150
MAXFLU=IQ	MP1	160
MAXNB=LQ	MP1	170
NMAX=20	MP1	180
WRITE (6,100) MAXKP,MAXZP,MAXTP,MAXRG,MAXZG,MAXTG,MAXNB	MP1	190
100 FORMAT(1H1,9X,19HT A C 3 D C O D E,5X,16HSTANDARD VERSION	MP1	200
2 ,/,30X,5HIQ = ,I2,4X,5HJQ = ,I2,4X,5HKQ = ,I2,4X,6HIGQ = ,I2,4X,	MP1	210
3,6HJQQ = ,I2,4X,6HKGQ = ,I2,4X,5HLQ = ,I3////)	MP1	220
C=====	MP1	230
IERROR(1)= 0	MP1	240
IERROR(2)= 0	MP1	250
ICOUNT=0	MP1	260
DO 10 I=1,25	MP1	270
SW(1)=.FALSE.	MP1	280
10 CONTINUE	MP1	290
C=====	MP1	300
C READ AND PRINT THE TITLE AND GEOMETRICAL DATA	MP1	310
CALL INPUT	MP1	320
C CHECK THE GEOMETRICAL DATA	MP1	330
CALL CHECK	MP1	340
C ASSIGN THE POINTS, BLOCKS AND GAPS	MP1	350
CALL POINTS	MP1	360
C CALCULATE THE CONSTANTS USED TO CORRECT FOR GAPS AND COOLANTS .	MP1	370
CALL CONSTA	MP1	380
C CALCULATE THE GEOMETRICAL PART OF THE RESISTANCES BETWEEN POINTS	MP1	390
CALL GEOMET	MP1	400
C PRINT THE LOCATION OF POINTS AND BOUNDARIES	MP1	410
CALL DPRINT	MP1	420
C READ THE INITIAL TEMPERATURE DISTRIBUTION	MP1	430
CALL INITEM	MP1	440
C READ AND PRINT THE COOLANT DATA	MP1	450

```

CALL FLOWCA MP1 460
C READ AND PRINT THE TIMESTEP SEQUENCE MP1 470
CALL TIME MP1 480
C READ AND PRINT THE GROUP OF SPECIAL FUNCTION PARAMETERS MP1 490
CALL EXTRA MP1 500
C READ THE PREVIOUS TEMPERATURE DISTRIBUTION, IF PRESENT MP1 510
CALL PRETEM MP1 520
CALL OTCARD MP1 530
C STOP EXECUTION IF ANY ERRORS HAVE BEEN FOUND MP1 540
IF (IERROR(1).NE.0) STOP MP1 550
C=====MP1 560
C INITIALIZE ARRAYS MP1 570
DO 20 IG=1,IGQ MP1 580
DO 20 J=1,JQ MP1 590
DO 20 K=1,KQ MP1 600
REML(IG,J,K) = 0.0 MP1 610
REMP(IG,J,K) = 0.0 MP1 620
20 CONTINUE MP1 630
DO 30 JG=1,JGQ MP1 640
DO 30 I=1,IQ MP1 650
DO 30 K=1,KQ MP1 660
ZEML(JG,I,K)=0.0 MP1 670
ZEMP(JG,I,K)=0.0 MP1 680
30 CONTINUE MP1 690
DO 40 KG=1,KGQ MP1 700
DO 40 I=1,IQ MP1 710
DO 40 J=1,JQ MP1 720
TEML(KG,I,J)=0.0 MP1 730
TEMP(KG,I,J)=0.0 MP1 740
40 CONTINUE MP1 750
C=====MP1 760
C SHOULD THE OUTPUT BE PUT ON A TAPE? -'NO,YES,- MP1 770
IF (.NOT.SW(3))RETURN MP1 780
C PUT THE INITIAL RECORD (TITLE,GRID INFORMATION) ON THE TAPE MP1 790
ILEN=1 MP1 800
WRITE (OUTTAP) ISHAPE,ILEN,IMAX,JMAX,KMAX,LMAX,MAXFLO,NRG,NZG,NTG,MP1 810
1(ZA(I),I=1,12),(RL(I),I=1,IMAX),(ZL(J),J=1,JMAX),(TL(K),K=1,KMAX),MP1 820
-(IGR(I),I=1,IMAX),(JGZ(J),J=1,JMAX),(KGT(K),K=1,KMAX), MP1 830
-(IL(L),IH(L),JL(L),JH(L),KL(L),KH(L),MB(L),L=1,LMAX) MP1 840
C=====MP1 850
RETURN MP1 860
C* *****MP1 870
C* *****MP1 880
C* *****MP1 890
SUBROUTINE OTCARD MP1 900
DIMENSION FORMT ( 8),FORM ( 8),FB ( 2) MP1 910

```

```

DATA FORMT      /6H( 10X,, 6*6H 12X,, 6HA6,A2)/      MP1  920
DATA FB        /6HF12.5,, 6HE12.5//                MP1  930
WRITE (6,90)(ZA(I),I=1,12)                          MP1  940
90 FORMAT(1H1,30X,12A6,///,40X,21HPROBLEM INPUT CARDS,///,10X, 80HMP1  950
1  0  1  1  2  2  3  3  4  4  5  5  6  6 MP1  960
2  7  7  8,/,10X, 80HMP1  970
3  5  0  5  0  5  0  5  0  5  0  5  0  5 MP1  980
4  0  5  0,/,10X, 80HMP1  990
5XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXMP1 1000
6XXXXXXXXXXXXXXXX,/) MP1 1010
IC=0 MP1 1020
100 IC=IC+1 MP1 1030
IF (IC.GT.NC) RETURN MP1 1040
ITYP=CARD(IC)+0.1 MP1 1050
IF (ITYP.EQ.0) GO TO 150 MP1 1060
IF (ITYP.EQ.7) GO TO 160 MP1 1070
DO 120 I=1,8 MP1 1080
FORM(I)=FORMT(I) MP1 1090
120 CONTINUE MP1 1100
DO 130 I=1,ITYP MP1 1110
FORM(I+1)=FB(I) MP1 1120
J=IC+I MP1 1130
IF (CARD(J).NE.0.0) GO TO 125 MP1 1140
CARD(J)=0.0 MP1 1150
GO TO 130 MP1 1160
125 IF (CARD(J) .LT. 0.0001 .OR. CARD(J) .GT. 9999.9999)FORM(I+1)=FB(J) MP1 1170
130 CONTINUE MP1 1180
140 I1=IC+1 MP1 1190
I2=IC+ITYP+2 MP1 1200
WRITE (6,FORM) (CARD(I),I=I1,I2) MP1 1210
GO TO 180 MP1 1220
150 WRITE (6,155) MP1 1230
155 FORMAT(82X,5HBLANK) MP1 1240
GO TO 100 MP1 1250
160 I1=IC+1 MP1 1260
I2=IC+14 MP1 1270
WRITE (6,170) (CARD(I),I=I1,I2) MP1 1280
170 FORMAT (10X,13A6,A2) MP1 1290
180 IC=12 MP1 1300
GO TO 100 MP1 1310
RETURN MP1 1320
C* *****MP1 1330
C* *****MP1 1340
END MP1 1350

```

```

SUBROUTINE INPUT
INCLUDE COMDIM
=====
C READ THE GEOMETRY DATA.
=====
C ERROR STOPS=
C INPUT1 THE GEOMETRY TYPE DESIRED HAS BEEN MISSPELLED.
C INPUT2 THE LOW RADIAL-X BLOCK BOUNDARY OF SOME BLOCK DOES
NOT COINCIDE WITH A RADIAL-X GRID LINE
C INPUT3 THE HIGH RADIAL-X BLOCK BOUNDARY OF SOME BLOCK DOES
NOT COINCIDE WITH A RADIAL-X GRID LINE
C INPUT4 THE LOW AXIAL-Z BLOCK BOUNDARY OF SOME BLOCK
DOES NOT COINCIDE WITH A AXIAL-Z GRID LINE
C INPUT5 THE HIGH AXIAL-Z BLOCK BOUNDARY OF SOME BLOCK
DOES NOT COINCIDE WITH AN AXIAL-Z GRID LINE
C INPUT6 THE LOW THETA-Y BLOCK BOUNDARY OF SOME BLOCK
DOES NOT COINCIDE WITH A THETA-Y GRID LINE
C INPUT7 THE HIGH THETA-Y BLOCK BOUNDARY OF SOME BLOCK
DOES NOT COINCIDE WITH A THETA-Y GRID LINE
=====
C
C=====
INTEGER BLANKS ,OPT ( 20),OPTION( 14)
DIMENSION FORM ( 11),FORMT ( 11),TYPE ( 10),GPRINT( 2),
1 MATGAP( 3),DIMGAP( 3),BTABLE(IQ,JQ,KQ)
EQUIVALENCE (BTABLE ,RCP )
DATA (GPRINT(I),I=1,2)
1 /6H,F10.4, 4H,I6./
DATA (FORM(I), I=1,11)
1 /1H(, 6HI4,2X,, 6H6F11.4, 6H,I6,3X, 4H,10X, 3H,6X,
2 4H,10X, 3H,6X, 4H,10X, 3H,6X, 1H)/
DATA (OPT(I), I = 1,20)
1 /4HDUMP, 5HPUNCH, 4HTAPE, 6HRESIST, 6HALL HE,
2 6HALL SU, 6HALL CO, 6HTIMEST, 6HALL DE, 6HSTEADY,
3 6HHEAT F, 6HSURFAC, 6HCONDUC, 6HRECTAN, 6HCYLIND,
4 6HCIRCUL, 6HRADIAL, 5HAXIAL, 5HTHETA,6HDECIMA/
DATA (TYPE(I),I=1,10)
- /6HRADIAL, 5HAXIAL, 5HTHETA, 4H X, 4H Y,
- 6H(INCHE, 2HS), 6H(DEGRE, 3HES), 4H 7/
DATA BLANKS /6H /
DATA BLANK /6H /
DATA STAR /5H* /
DATA GAP /5H# /
DATA DOT /5H. /
=====
C PNAME(4) = TYPE(6)
=====

```

```

INPU 10
INPU 20
INPU 30
INPU 40
INPU 50
INPU 60
INPU 70
INPU 80
INPU 90
INPU 100
INPU 110
INPU 120
INPU 130
INPU 140
INPU 150
INPU 160
INPU 170
INPU 180
INPU 190
INPU 200
INPU 210
INPU 220
INPU 230
INPU 240
INPU 250
INPU 260
INPU 270
INPU 280
INPU 290
INPU 300
INPU 310
INPU 320
INPU 330
INPU 340
INPU 350
INPU 360
INPU 370
INPU 380
INPU 390
INPU 400
INPU 410
INPU 420
INPU 430
INPU 440
INPU 450

```

```

PNAME(5) = TYPE(7) INPU 460
PNAME(6) = TYPE(6) INPU 470
PNAME(7) = TYPE(7) INPU 480
NC=0 INPU 490
C=====INPU 500
C READ AND PRINT THE TITLE INPU 510
CALL TITLE INPU 520
C=====INPU 530
C READ THE NEXT OPTION INPU 540
1 READ (5,103) (OPTION(I),I=1,14) INPU 550
103 FORMAT(13A6,A2) INPU 560
WRITE(6,100) (OPTION(I), I= 1,14) INPU 570
100 FORMAT (1H0,19X,12A6) INPU 580
C HAVE ALL OPTIONS BEEN READ? -'YES,NO'- INPU 590
IF(OPTION(2).EQ.BLANKS)GO TO 4 INPU 600
CALL ACARD(OPTION) INPU 610
C/ ASSIGN THE OPTION -'OK,ERROR'- INPU 620
DO 2 I=1,20 INPU 630
IF(OPTION(2).NE.OPT(I))GO TO 2 INPU 640
SW(1)=.TRUE. INPU 650
GO TO 1 INPU 660
2 CONTINUE INPU 670
CE *****INPU 680
C OPTION COULD NOT BE FOUND, PRINT A WARNING INPU 690
WRITE(6,99)OPTION(2) INPU 700
99 FORMAT(6H1OPTION ,A6,18H COULD NOT BE READ) INPU 710
GO TO 1 INPU 720
C=====INPU 730
4 NC=NC+1 INPU 740
CARD(NC)=0.0 INPU 750
C SPECIFY THE PRINT FORMAT INPU 760
NPRINT=3 INPU 770
IF(SW(17)) NPRINT=1 INPU 780
IF(SW(18)) NPRINT=2 INPU 790
IF(SW(19)) NPRINT=3 INPU 800
C=====INPU 810
C IS IT A RECTANGULAR GEOMETRY? -'YES,NO'- INPU 820
IF(SW(14))GO TO 5 INPU 830
C IS IT A CYLINDRICAL GEOMETRY? -'YES,NO'- INPU 840
IF(SW(15))GO TO 3 INPU 850
C THE TYPE OF GEOMETRY IS NOT SPECIFIED CORRECTLY INPU 860
CALL ERROR(6HINPUT1) INPU 870
C=====INPU 880
C SET THE PROBLEM UP FOR A CYLINDRICAL GEOMETRY INPU 890
3 SCALE=57.295795 INPU 900
ISHAPE=0 INPU 910

```



```

PNAME(1) = TYPE(1) INPU 920
PNAME(2) = TYPE(2) INPU 930
PNAME(3)=TYPE(3) INPU 940
PNAME(4)=TYPE(8) INPU 950
PNAME(5)=TYPE(9) INPU 960
GO TO 10 INPU 970
C=====INPU 980
C SET THE PROBLEM UP FOR A RECTANGULAR GEOMETRY INPU 990
  5 SCALE=12.0 INPU1000
  ISHAPE=1 INPU1010
  PNAME(1) = TYPE(4) INPU1020
  PNAME(2) = TYPE(5) INPU1030
  PNAME(3)=TYPE(10) INPU1040
C=====INPU1050
CB READ AND PROCESS THE GRID DESCRIPTIONS INPU1060
  10 IM=MAXRP-1 INPU1070
C READ THE INPUT DESCRIBING THE RADIAL-X GRID LINES INPU1080
  CALL GRID (RL,IM) INPU1090
  JM= MAXZP-1 INPU1100
C READ THE INPUT DESCRIBING THE AXIAL-Z GRID LINES INPU1110
  CALL GRID (ZL,JM) INPU1120
  KM= MAXTP-1 INPU1130
C READ THE INPUT DESCRIBING THE THETA-Y GRID LINES INPU1140
  CALL GRID (TL,KM) INPU1150
C=====INPU1160
C ASSIGN THE GRID LIMITS INPU1170
  IMAX=IM+1 INPU1180
  JMAX=JM+1 INPU1190
  KMAX=KM+1 INPU1200
  IM1=IM-1 INPU1210
  JM1=JM-1 INPU1220
  KM1=KM-1 INPU1230
C=====INPU1240
C CHANGE A ZERO INNER RADIUS TO A SMALL BUT FINITE ONE INPU1250
  IF (RL(1) .LE. 0.0) RL(1) = 1.0E-5 INPU1260
C CHANGE THE RADIAL-X GRID LINE DATA TO THE CORRECT UNITS INPU1270
  DO 20 I=1,IM INPU1280
  RL(I)=RL(I)/12.0 INPU1290
  20 CONTINUE INPU1300
C=====INPU1310
C CHANGE THE AXIAL-Z GRID LINE DATA TO THE CORRECT UNITS INPU1320
  DO 30 J=1,JM INPU1330
  ZL(J)=ZL(J)/12.0 INPU1340
  30 CONTINUE INPU1350
C=====INPU1360
C CHANGE THE THETA-Y GRID LINE DATA TO THE CORRECT UNITS INPU1370

```

```

DO 40 K=1,KM                                INPU1380
TL(K)=TL(K)/SCALE                            INPU1390
40 CONTINUE                                  INPU1400
C=====INPU1410
C PRINT THE HEADING OF THE BLOCK PRINT      INPU1420
WRITE (6,105) (ZA(I), I= 1,12), PNAME(1), PNAME(1), PNAME(2), INPU1430
-PNAME(2), PNAME(3), PNAME(3), PNAME(1), PNAME(2), PNAME(3), INPU1440
-PNAME(4), PNAME(5), PNAME(4), PNAME(5), PNAME(4), PNAME(5) INPU1450
105 FORMAT (1H1,30X,12A6,///,50X,18HBLOCK DESCRIPTION,///,2X,5HBLOCK, INPU1460
-3X,66H. . . . . BOUNDARIES . . . . . INPU1470
- . . . . .9X,44H. . . . . GAPS . . . . ./1X, INPU1480
-6HNUMBER/8X,4HLOW ,A6,6H HIGH ,A6,5H LOW ,A6,5HHIGH ,A6,4HLOW , INPU1490
-A6,5HHIGH ,A6,6HMATERIAL,4X,A6,11H MATERIAL ,A6,10HMATERIAL ,A6, INPU1500
-11HMATERIAL/9X,8H(INCHES),3X,8H(INCHES),4X,8H(INCHES),2X,8H(INCHES) INPU1510
-,3X,A6,A3,1X,A6,A3,13X,8H(INCHES),8X,8H(INCHES),8X,A6,A3//) INPU1520
C=====INPU1530
CB READ AND PRINT THE BLOCK DESCRIPTIONS    INPU1540
K=0                                          INPU1550
C COUNT THE BLOCKS                          INPU1560
50 K=K+1                                    INPU1570
C READ THE MATERIAL NUMBER                  INPU1580
READ (5,101) X,Y,FLAG1,FLAG2              INPU1590
101 FORMAT(2E12.4,48X,A6,A2)              INPU1600
C HAVE ALL THE BLOCK CARDS BEEN READ? -YES,NO,- INPU1610
IF (X.EQ.0.0) GO TO 60                    INPU1620
CALL FCARD(2.0,X,Y,FLAG1,FLAG2,DUM,DUM,DUM,DUM) INPU1630
C IS IT A COOLANT? -NO,YES,-              INPU1640
IF (X.GT.0.0) GO TO 55                   INPU1650
C ASSIGN THE FLOW DIRECTION OF THE COOLANT INPU1660
N=-(X-0.1)                                INPU1670
IPATH(N)= Y                                INPU1680
C ASSIGN THE COOLANT OR MATERIAL NUMBER    INPU1690
55 MB(K)=X                                  INPU1700
C=====INPU1710
C READ THE BLOCK DIMENSIONS                 INPU1720
READ (5,104) RBL(K),RBH(K),ZBL(K),ZBH(K),TBL(K),TBH(K),FLAG1,FLAG2 INPU1730
104 FORMAT(2E12.4,A6,A2)                  INPU1740
CALL FCARD(6.0,RBL(K),RBH(K),ZBL(K),ZBH(K),TBL(K),TBH(K),FLAG1, INPU1750
XFLAG2)                                    INPU1760
C=====INPU1770
C READ THE GAP PROPERTIES                   INPU1780
READ (5,104) RDG(K),X,ZDG(K),Y,TDG(K),Z,FLAG1,FLAG2 INPU1790
CALL FCARD(6.0,RDG(K),X,ZDG(K),Y,TDG(K),Z,FLAG1,FLAG2) INPU1800
MGR(K)=X                                    INPU1810
MGZ(K)=Y                                    INPU1820
MGT(K)=Z                                    INPU1830

```

```

=====INPU1840
C PRINT THE BLOCK DEFINITION INPU1850
  IND=0 INPU1860
  DO 240 I=1,11 INPU1870
  FORMT(I)=FORM(I) INPU18A0
240 CONTINUE INPU1890
  DO 308 I=1,3 INPU1900
  GO TO (301,302,303),I INPU1910
301 IF (X.LE.0.0) GO TO 308 INPU1920
  NG=NGK(K) INPU1930
  GG=KGK(K) INPU1940
  GO TO 305 INPU1950
302 IF (Y.LE. 0.0) GO TO 308 INPU1960
  MG=MGZ(K) INPU1970
  GG=ZKG(K) INPU19A0
  GO TO 305 INPU1990
303 IF (Z.LE. 0.0) GO TO 308 INPU2000
  NG=NGT(K) INPU2010
  GG=TKG(K) INPU2020
305 IND=IND+1 INPU2030
  MATGAP(IND)=MG INPU2040
  DIMGAP(IND)=GG INPU2050
  II=I+2 INPU2060
  FORMT(II+3)=GPRINT(1) INPU2070
  FORMT(II+4)=GPRINT(2) INPU20A0
308 CONTINUE INPU2090
  IF (IND.EQ. 0) GO TO 311 INPU2100
  WRITE (6,FORMAT)K,RBL(K),RBH(K),ZBL(K),ZBH(K),TBL(K),TBH(K),MB(K), INPU2110
  1(DIMGAP(I),MATGAP(I),I=1,IND) INPU2120
  GO TO 50 INPU2130
311 WRITE (6,FORMAT)K,RBL(K),RBH(K),ZBL(K),ZBH(K),TBL(K),TBH(K),MB(K) INPU2140
  GO TO 50 INPU2150
=====INPU2160
C ASSIGN THE NUMBER OF BLOCKS INPU2170
  DO NC=NC+1 INPU21A0
  CARU(NC)=0.0 INPU2190
  LMAX=K-1 INPU2200
C/ CONVERT THE BLOCK AND GAP DIMENSIONS TO THEIR CORRECT UNITS INPU2210
  DO 70 K=1,LMAX INPU2220
  IF (RBL(K) .LE. 0.0) RBL(K) = 1.0E-5 INPU2230
  RBL(K)=RBL(K)/12.0 INPU2240
  IF (RBH(K) .LE. 0.0) RBH(K) = 1.0E-5 INPU2250
  RBH(K)=RBH(K)/12.0 INPU2260
  ZBL(K)=ZBL(K)/12.0 INPU2270
  ZBH(K)=ZBH(K)/12.0 INPU22A0
  TBL(K)=TBL(K)/SCALE INPU2290

```

	TBH(K)=TBH(K)/SCALE	INPU2300
	RDG(K)=RDG(K)/12.0	INPU2310
	ZDG(K)=ZDG(K)/12.0	INPU2320
	TDG(K)=TDG(K)/SCALE	INPU2330
70	CONTINUE	INPU2340
CE	*****	INPU2350
C	TEST THAT BLOCK BOUNDARIES COINCIDE WITH GRID LINES	INPU2360
	DO 261 I=1,IMAX	INPU2370
	DO 261 J=1,JMAX	INPU2380
	DO 261 K=1,KMAX	INPU2390
	BTABLE(I,J,K)=BLANK	INPU2400
261	CONTINUE	INPU2410
	DO 400 L=1,LMAX	INPU2420
	DO 270 ILG=1,IM	INPU2430
	IF (RBL(L).GE.0.99999*RL(ILG).AND.RBL(L).LE.1.00001*RL(ILG)) GO TO	INPU2440
	1 280	INPU2450
270	CONTINUE	INPU2460
	CALL ERROR (6HINPUT2)	INPU2470
280	DO 290 IHG=1,IM	INPU2480
	IF (RBH(L).GE.0.99999*RL(IHG).AND.RBH(L).LE.1.00001*RL(IHG)) GO TO	INPU2490
	1 300	INPU2500
290	CONTINUE	INPU2510
	CALL ERROR (6HINPUT3)	INPU2520
300	DO 310 JLG=1,JM	INPU2530
	IF (ZBL(L).GE.0.99999*ZL(JLG).AND.ZBL(L).LE.1.00001*ZL(JLG)) GO TO	INPU2540
	1 320	INPU2550
310	CONTINUE	INPU2560
	CALL ERROR (6HINPUT4)	INPU2570
320	DO 330 JHG=1,JM	INPU2580
	IF (ZBH(L).GE.0.99999*ZL(JHG).AND.ZBH(L).LE.1.00001*ZL(JHG)) GO TO	INPU2590
	1 340	INPU2600
330	CONTINUE	INPU2610
	CALL ERROR (6HINPUT5)	INPU2620
340	DO 350 KLG=1,KM	INPU2630
	IF (TBL(L).GE.0.99999*TL(KLG).AND.TBL(L).LE.1.00001*TL(KLG))	INPU2640
	XGO TO 360	INPU2650
350	CONTINUE	INPU2660
	CALL ERROR (6HINPUT6)	INPU2670
360	DO 370 KHG=1,KM	INPU2680
	IF (TBH(L).GE.0.99999*TL(KHG).AND.TBH(L).LE.1.00001*TL(KHG))	INPU2690
	XGO TO 331	INPU2700
370	CONTINUE	INPU2710
	CALL ERROR (6HINPUT7)	INPU2720
331	IF (MB(L).GT.0) GO TO 335	INPU2730
	DO 332 J=JLG,JHG	INPU2740
	DO 332 K=KLG,KHG	INPU2750

```

BTABLE(ILG,J,K)=STAR
BTABLE(IHG,J,K)=STAR
332 CONTINUE
DO 333 I=ILG,IHG
DO 333 K=KLG,KHG
BTABLE(I,JLG,K)=STAR
BTABLE(I,JHG,K)=STAR
333 CONTINUE
DO 334 I=ILG,IHG
DO 334 J=JLG,JHG
BTABLE(I,J,KLG)=STAR
BTABLE(I,J,KHG)=STAR
334 CONTINUE
GO TO 400
335 RHS=DOT
ZHS=DOT
THS=DOT
IF (RDG(K).GT.0.0) RHS=GAP
IF (ZDG(K).GT.0.0) ZHS=GAP
IF (TDG(K).GT.0.0) THS=DOT
DO 336 J=JLG,JHG
DO 336 K=KLG,KHG
IF (BTABLE(IHG,J,K).NE.STAR .AND. BTABLE(IHG,J,K).NE.GAP)
XBTABLE(IHG,J,K)=RHS
336 CONTINUE
DO 337 I=ILG,IHG
DO 337 K=KLG,KHG
IF (BTABLE(I,JHG,K).NE.STAR .AND. BTABLE(I,JHG,K).NE.GAP)
XBTABLE(I,JHG,K)=ZHS
337 CONTINUE
DO 338 I=ILG,IHG
DO 338 J=JLG,JHG
IF (BTABLE(I,J,KHG).NE.STAR .AND. BTABLE(I,J,KHG).NE.GAP)
XBTABLE(I,J,KHG)=THS
338 CONTINUE
400 CONTINUE
WRITE (6,380) (ZA(I),I=1,12)
380 FORMAT(1H1,30X,12A6,///,50X,17HBOUNDARY OVERLAY,///,55X,29H* WHERE
1E COOLANTS ARE PRESENT,/,55X,25H* WHERE GAPS ARE PRESENT,/,55X,37
2H. WHERE GAPS OR COOLANTS NOT PRESENT,/)
CALL BARRAY
C=
RETURN
C=
CE *****
CE *****

```

```

INPU2760
INPU2770
INPU2780
INPU2790
INPU2800
INPU2810
INPU2820
INPU2830
INPU2840
INPU2850
INPU2860
INPU2870
INPU2880
INPU2890
INPU2900
INPU2910
INPU2920
INPU2930
INPU2940
INPU2950
INPU2960
INPU2970
INPU2980
INPU2990
INPU3000
INPU3010
INPU3020
INPU3030
INPU3040
INPU3050
INPU3060
INPU3070
INPU3080
INPU3090
INPU3100
INPU3110
INPU3120
INPU3130
INPU3140
INPU3150
INPU3160
INPU3170
INPU3180
INPU3190
INPU3200
INPU3210

```

```

CE *****INPU3220
SUBROUTINE TITLE INPU3230
C=====INPU3240
C READ AND PRINT TITLE CARDS UNTIL A BLANK CARD INPU3250
C IS ENCOUNTERED. PRESERVE THE FIRST LINE INPU3260
C=====INPU3270
INTEGER ZZ ( 14) INPU3280
C READ, RECORD AND PRINT THE FIRST TITLE CARD INPU3290
READ (5,140) (ZA(I),I=1,14) INPU3300
CALL ACARD(ZA) INPU3310
WRITE (6,150) (ZA(I),I=1,12) INPU3320
C READ THE NEXT TITLE CARD INPU3330
100 READ (5,140) (ZZ(I),I=1,14) INPU3340
C/ HAVE ALL TITLE CARDS BEEN READ? NO,YES=- INPU3350
DO 110 I=1,12 INPU3360
IF (ZZ(I).NE.BLANKS) GO TO 130 INPU3370
110 CONTINUE INPU3380
NC=NC+1 INPU3390
CARD(NC)=0.0 INPU3400
WRITE (6,120) INPU3410
120 FORMAT (1H0,15X,7HOPTIONS) INPU3420
CE *****INPU3430
RETURN INPU3440
C=====INPU3450
C PRINT THE TITLE CARD INPU3460
130 WRITE (6,150) (ZZ(I),I=1,12) INPU3470
CALL ACARD(ZZ) INPU3480
GO TO 100 INPU3490
C=====INPU3500
140 FORMAT (13A6,A2) INPU3510
150 FORMAT (10X,12A6) INPU3520
C=====INPU3530
CE *****INPU3540
CE *****INPU3550
CE *****INPU3560
SUBROUTINE GRID(RL,IM1) INPU3570
C=====INPU3580
C READ ONE SET OF GRID DATA CARDS INPU3590
C=====INPU3600
C ERROR STOPS= INPU3610
C GRID1 THERE ARE MORE GRIDLINES THEN ALLOWED IN ONE OF INPU3620
C THE DIMENSIONS. A BLANK CARD HAS BEEN LEFT OUT. INPU3630
C=====INPU3640
C=====INPU3650
INTEGER IM INPU3660
DIMENSION RL ( 8),RL1 ( 1) INPU3670

```

```

C=====INPU36A0
      ILS= 1                                INPU3690
      IHS= 6                                INPU3700
C      READ THE NEXT GRID DATA CARD        INPU3710
100 READ (5,140) (RL(I),I=1,8)            INPU3720
C      HAVE ALL GRID DATA CARDS BEEN READ? -'YES,NO'- INPU3730
      IF (RL(1).EQ.0.0.AND.RL(2).EQ.0.0) GO TO 120 INPU3740
      CALL FCARD(6.0,RL(1),RL(2),RL(3),RL(4),RL(5),RL(6),RL(7),RL(8)) INPU3750
C=====INPU3760
      IM= MINO(IHS,IM1)                     INPU3770
C      ARE THERE TOO MANY GRID DATA CARDS ? -'YES,NO'- INPU37A0
C      PRINT ERROR GRID 1                   INPU3790
      IF (ILS.GT.IM) CALL ERROR(6H GRID1)    INPU3800
C      RECORD THE CONTENTS OF THE CURRENT GRID DATA CARD INPU3810
      DO 110 I= ILS,IM                      INPU3820
      J= I+1-ILS                            INPU3830
      RL1(I)= RL(J)                         INPU3840
110 CONTINUE                               INPU3850
C      PREPARE FOR THE READING OF THE NEXT GRID DATA CARD INPU3860
      ILS= IHS+1                            INPU3870
      IHS= IHS+6                            INPU38A0
      GO TO 100                              INPU3890
C=====INPU3900
C/      DETERMINE THE NUMBER OF GRIDLINES READ INPU3910
120 NC=IC+1                                INPU3920
      CARL(NC)=0.00                         INPU3930
      IM=ILS                                INPU3940
130 IM=IM-1                                INPU3950
      IF (RL1(IM).LE.0.0) GO TO 130         INPU3960
      IM1= IM                               INPU3970
CE *****INPU39A0
      RETURN                                 INPU3990
C=====INPU4000
140 FORMAT (6E12.4,A6,A2)                  INPU4010
C=====INPU4020
CE *****INPU4030
CE *****INPU4040
CE *****INPU4050
      SUBROUTINE BARRAY                     INPU4060
C=====INPU4070
      DIMENSION FORMN1( 3),FORMN2( 3),FORMA1( 3),FORMA2( 3), INPU40A0
1      DIMENSION FORMN ( 3),FORMA ( 3)      INPU4090
      DIMENSION NUMBER( 50)                INPU4100
      DATA (FORMN1(I),I=1,3) /5H(1H0,, 3H6X,, 5H25I5)/ INPU4110
      DATA (FORMN2(I),I=1,3) /5H(1H0,, 3H1X,, 5H26I5)/ INPU4120
      DATA (FORMA1(I),I=1,3) /4H(I4,, 3H7X,, 5H24A5)/ INPU4130

```

```

DATA (FORMA2(I),I=1,3) /4H(I4,, 3H2X,, 5H25A5)/ INPU4140
DATA NUMBER /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19, INPU4150
- /20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35, INPU4160
- /36,37,38,39,40,41,42,43,44,45,46,47,48,49,50/ INPU4170
C=====INPU4180
C WRITE THE PROPER HEADING INPU4190
GO TO (10,20,30), NPRINT INPU4200
10 WRITE (6,1) PNAME(3), PNAME(2) INPU4210
1 FORMAT (1H0,42X,4HTHE ,A6,27H(K) DIRECTION IS HORIZONTAL/43X, INPU4220
-4HTHE ,A6,25H(J) DIRECTION IS VERTICAL) INPU4230
MM=JM INPU4240
N=KN INPU4250
GO TO 40 INPU4260
20 WRITE (6,2) PNAME(1), PNAME(3) INPU4270
2 FORMAT (1H0,42X,4HTHE ,A6,27H(I) DIRECTION IS HORIZONTAL/43X, INPU4280
-4HTHE ,A6,25H(K) DIRECTION IS VERTICAL) INPU4290
MM=JM INPU4300
N=IN INPU4310
GO TO 40 INPU4320
30 WRITE (6,3) PNAME(1), PNAME(2) INPU4330
3 FORMAT (1H0,42X,4HTHE ,A6,27H(I) DIRECTION IS HORIZONTAL/43X, INPU4340
-4HTHE ,A6,25H(J) DIRECTION IS VERTICAL) INPU4350
MM=KM INPU4360
N=IM INPU4370
40 DO 600 M=1,MM INPU4380
NUM1=1 INPU4390
NUM2=24 INPU4400
IF (NUM2 .GE. N) NUM2=N INPU4410
DO210 I=1,3 INPU4420
FORMN(I)=FORMN1(I) INPU4430
FORMA(I)=FORMA1(I) INPU4440
210 CONTINUE INPU4450
WRITE (6,4) PNAME(NPRINT),M INPU4460
4 FORMAT(1H0,/,48X,A6,2H (,I2,7H) PLANE) INPU4470
150 WRITE (6,FORMN) (NUMBER(L),L=NUM1,NUM2) INPU4480
WRITE (6,170) INPU4490
170 FORMAT (1H0) INPU4500
GO TO (200,300,400),NPRINT INPU4510
C=====INPU4520
C OUTPUT PRINT IN RADIAL-X DIRECTION, THETA-Y DIRECTION HORIZONTAL, INPU4530
C AXIAL-Z DIRECTION VERTICAL INPU4540
C=====INPU4550
200 DO 420 J=1,JM INPU4560
420 WRITE (6,FORMA) J,(BTABLE(M,J,K),K=NUM1,NUM2) INPU4570
IF (NUM2 .LT. N) GO TO 100 INPU4580
GO TO 600 INPU4590

```



```

C=====INPU4600
C   OUTPUT PRINT IN AXIAL-Z DIRECTION, RADIAL-X DIRECTION HORIZONTAL, INPU4610
C   THETA-Y DIRECTION VERTICAL INPU4620
C=====INPU4630
  300 DO 320 K=1,KM INPU4640
  320 WRITE (6,FORMA) K, (BTABLE(I,M,K) , I=NUM1,NUM2) INPU4650
      IF(NUM2 .LT. N) GO TO 100 INPU4660
      GO TO 600 INPU4670
C=====INPU4680
C   OUTPUT PRINT IN THETA-Y DIRECTION, RADIAL-X DIRECTION HORIZONTAL, INPU4690
C   AXIAL-Z DIRECTION VERTICAL INPU4700
C=====INPU4710
  400 DO 230 J=1,JM INPU4720
  230 WRITE (6,FORMA) J, (BTABLE(I,J,M),I=NUM1,NUM2) INPU4730
      IF(NUM2 .LT. N) GO TO 100 INPU4740
      GO TO 600 INPU4750
  100 NUM1=NUM2+1 INPU4760
      NUM2=NUM2 + 25 INPU4770
      IF (NUM2 .GE. N) NUM2=N INPU4780
      DO 110 I=1,3 INPU4790
      FORM1(I)=FORM2(I) INPU4800
      FORMA(I)=FORMA2(I) INPU4810
  110 CONTINUE INPU4820
      GO TO 150 INPU4830
  600 CONTINUE INPU4840
      RETURN INPU4850
C=====INPU4860
SUBROUTINE ACARD(AC) INPU4870
C=====INPU4880
DIMENSION AC ( 14) INPU4890
C=====8INPU4900
NC=NC+1 INPU4910
CARD(NC)=7.0 INPU4920
DO 100 I=1,14 INPU4930
NC=NC+1 INPU4940
CARD(NC)=AC(I) INPU4950
  100 CONTINUE INPU4960
      RETURN INPU4970
CE *****INPU4980
CE *****INPU4990
      END INPU5000

```

```

SUBROUTINE ERROR(IB)
INCLUDE COMDIM
=====
C=====
C=====
C=====
C=====
DIMENSION IB(2), NOCNT(2)
DATA (NOCNT(I),I=1,2) / 5HTIME1, 5HTIME2 /
DATA NUMBER/5/
=====
C=====
IERROK(1) = IB(1)
IERROK(2) = IB(2)
IF((IERROK(1).EQ.NOCNT(1).OR.IERROR(1).EQ.NOCNT(2)).AND.
1 SW(10)) GO TO 90
NUMBER=NUMBER-1
C RECORD THE ERROR
90 WRITE(6,100) IERROR
100 FORMAT(10H ERROR AT ,2A6)
IF (NUMBER .LE. 0) STOP
RETURN
C=====
END
=====

```

	ERRO 10
	ERRO 20
	ERRO 30
	ERRO 40
	ERRO 50
	ERRO 60
	ERRO 70
	ERRO 80
	ERRO 90
	ERRO 100
	ERRO 110
	ERRO 120
	ERRO 130
	ERRO 140
	ERRO 150
	ERRO 160
	ERRO 170
	ERRO 180
	ERRO 190
	ERRO 200
	ERRO 210
	ERRO 220

	SUBROUTINE FCARD (ANUM,C1,C2,C3,C4,C5,C6,C7,C8)	FCAR 10
	INCLUDE COMDIM	FCAR 20
C=	=====	FCAR 30
C=	=====	FCAR 40
	DIMENSION C (8)	FCAR 50
C=	=====	FCAR 60
	C(1)=C1	FCAR 70
	C(2)=C2	FCAR 80
	C(3)=C3	FCAR 90
	C(4)=C4	FCAR 100
	C(5)=C5	FCAR 110
	C(6)=C6	FCAR 120
	C(7)=C7	FCAR 130
	C(8)=C8	FCAR 140
	NC=NC+1	FCAR 150
	CARD(NC)=ANUM	FCAR 160
	I1=ANUM+0.1	FCAR 170
	I1=I1+2	FCAR 180
	DO 100 I=1,II	FCAR 190
	NC=NC+1	FCAR 200
	CARD(NC)=C(I)	FCAR 210
100	CONTINUE	FCAR 220
	RETURN	FCAR 230
	END	FCAR 240

```

SUBROUTINE CHECK                                CHEC 10
INCLUDE      COMDIM                              CHEC 20
C=====CHEC 30
C   CHECK THE GEOMETRY DATA                    CHEC 40
C=====CHEC 50
C   ERROR STOPS=                                CHEC 60
C   CHECK1   THERE ARE TOO MANY RADIAL-X POINTS AND GRID LINES CHEC 70
C   CHECK2   THERE ARE TOO MANY AXIAL-Z POINTS AND GRID LINES CHEC 80
C   CHECK3   THERE ARE TOO MANY THETA-Y POINTS AND GRID LINES CHEC 90
C   CHECK4   THE RADIAL-X GRID DATA IS OUT OF ORDER           CHEC 100
C   CHECK5   THE AXIAL-Z GRID DATA IS OUT OF ORDER           CHEC 110
C   CHECK6   THE THETA-Y GRID DATA IS OUT OF ORDER           CHEC 120
C   CHECK7   THERE ARE TOO MANY BLOCKS.                       CHEC 130
C   CHECK8   THE LOW RADIAL-X BOUNDARY IS LARGER THAN THE HIGH CHEC 140
C             RADIAL-X BOUNDARY FOR SOME BLOCK                 CHEC 150
C   CHECK9   THE LOW AXIAL-Z BOUNDARY IS GREATER THAN THE     CHEC 160
C             HIGH AXIAL-Z BOUNDARY FOR SOME BLOCK             CHEC 170
C   CHECK10  THE LOW THETA-Y BOUNDARY IS GREATER THAN THE     CHEC 180
C             HIGH THETA-Y BOUNDARY FOR SOME BLOCK             CHEC 190
C   CHECK11  THE MATERIAL NUMBER FOR A BLOCK IS LARGER THAN THE CHEC 200
C             MAXIMUM NUMBER OF MATERIALS AND COOLANTS ALLOWED. CHEC 210
C   CHECK12  A RADIAL-X GAP MATERIAL NUMBER IS TOO HIGH       CHEC 220
C   CHECK13  AN AXIAL-Z GAP MATERIAL NUMBER IS TOO HIGH       CHEC 230
C   CHECK14  AN THETA-Y GAP MATERIAL NUMBER IS TOO HIGH       CHEC 240
C=====CHEC 250
C=====CHEC 260
C   ARE THERE TOO MANY RADIAL-X GRIDLINES? -'YES,NO'-        CHEC 270
C   PRINT ERROR CHECK1                                       CHEC 280
C   IF (IMAX.GT.VMAXRP) CALL ERROR(6HCHECK1)                  CHEC 290
C   ARE THERE TOO MANY AXIAL-Z GRIDLINES? -'YES,NO'-        CHEC 300
C   PRINT ERROR CHECK2                                       CHEC 310
C   IF (JMAX.GT.VMAXZP) CALL ERROR(6HCHECK2)                  CHEC 320
C   ARE THERE TOO MANY THETA-Y GRIDLINES? -'YES,NO'-        CHEC 330
C   PRINT ERROR CHECK3                                       CHEC 340
C   IF (KMAX.GT.VMAXTP) CALL ERROR(6HCHECK3)                  CHEC 350
C=====CHEC 360
C/  CHECK TO SEE THAT THE RADIAL-X GRID LINES ARE IN AN ASCENDING . CHEC 370
C   -ORDER                                                    CHEC 380
C   DO 10 I=2,IM                                             CHEC 390
C     IF (RL(I).LE.RL(I-1)) CALL ERROR(6HCHECK4)              CHEC 400
C 10 CONTINUE                                                CHEC 410
CE *****CHEC 420
C/  CHECK TO SEE THAT THE AXIAL-Z GRIDLINES ARE IN AN ASCENDING ORDER CHEC 430
C   DO 20 J=2,JM                                             CHEC 440
C     IF (ZL(J).LE.ZL(J-1)) CALL ERROR(6HCHECK5)              CHEC 450

```

```

20 CONTINUE CHEC 460
CE *****CHEC 470
C/ CHECK TO SEE THAT THE THETA-Y GRIDLINES ARE IN AN ASCENDING ORDER CHEC 480
DO 30 K=2,KM CHEC 490
IF (TL(K).LE.TL(K-1)) CALL ERROR(6HCHECK6) CHEC 500
30 CONTINUE CHEC 510
CE *****CHEC 520
C ARE THERE TOO MANY BLOCKS? -'YES,NO'- CHEC 530
C PRINT ERROR CHECK7 CHEC 540
IF (LMAX.GT.MAXNB) CALL ERROR (6HCHECK7) CHEC 550
=====CHEC 560
CB CHECK THE BLOCK GEOMETRY FOR INTERNAL CONSISTENCY CHEC 570
C CHECK EVERY BLOCK IN TURN CHEC 580
DO 40 L=1,LMAX CHEC 590
C ARE THE RADIAL-X BOUNDARIES IN ORDER? -'NO,YES'- CHEC 600
C PRINT ERROR CHECK8 CHEC 610
IF (RBL(L).GT.RBH(L)) CALL ERROR(6HCHECK8) CHEC 620
C ARE THE AXIAL-Z BOUNDARIES IN ORDER? -'NO,YES'- CHEC 630
C PRINT ERROR CHECK9 CHEC 640
IF (ZBL(L).GT.ZBH(L)) CALL ERROR(6HCHECK9) CHEC 650
C ARE THE THETA-Y BOUNDARIES IN ORDER? -'NO,YES'- CHEC 660
C PRINT ERROR CHECK10 CHEC 670
IF (TBL(L).GT.TBH(L)) CALL ERROR(7HCHECK10) CHEC 680
C IS THE MATERIAL NUMBER A VALID ONE? -'NO,YES'- CHEC 690
C PRINT ERROR CHECK11 CHEC 700
IF (IABS(MB(L)).GT.MAXMAT) CALL ERROR(7HCHECK11) CHEC 710
C IS THE MATERIAL NUMBER OF THE RADIAL-X GAP A VALID ONE? -'NO,YES'-CHEC 720
C PRINT ERROR CHECK12 CHEC 730
IF (MBR(L).GT.MAXMAT) CALL ERROR(7HCHECK12) CHEC 740
C IS THE MATERIAL NUMBER OF THE AXIAL-Z GAP A VALID ONE? -'NO,YES'- CHEC 750
C PRINT ERROR CHECK13 CHEC 760
IF (MBZ(L).GT.MAXMAT) CALL ERROR(7HCHECK13) CHEC 770
C IS THE MATERIAL NUMBER OF THE THETA-Y GAP A VALID ONE? -'NO,YES'- CHEC 780
C PRINT ERROR CHECK14 CHEC 790
IF (MBT(L).GT.MAXMAT) CALL ERROR(7HCHECK14) CHEC 800
C HAVE ALL BLOCKS BEEN CHECKED ? -'YES,NO'- CHEC 810
40 CONTINUE CHEC 820
CE *****CHEC 830
RETURN CHEC 940
=====CHEC 850
END CHEC 860

```

```

SUBROUTINE POINTS                                POIN  10
INCLUDE      COMDIM                              POIN  20
=====POIN  30
C  ASSIGN THE POINTS, BLOCKS AND GAPS            POIN  40
C  =====POIN  50
C  ERROR STOPS=                                  POIN  60
C  POINTS1   A PART OF THE SYSTEM WAS NOT DESCRIBED BY ANY BLOCK. POIN  70
C  POINTS2   THERE ARE TOO MANY RADIAL-X GAPS.      POIN  80
C  POINTS3   THERE ARE TOO MANY AXIAL-Z GAPS.      POIN  90
C  POINTS4   THERE ARE TOO MANY THETA-Y GAPS.      POIN 100
C  POINTS5   A PART OF THE SYSTEM HAS BEEN DESCRIBED BY MORE THAN POIN 110
C            ONE BLOCK.                             POIN 120
C  POINTS6   NO FLOW DIRECTION HAS BEEN ASSIGNED FOR SOME COOLANT. POIN 130
C  POINTS7   1. AN EXTERNAL COOLANT IS FLOWING INTO A RADIAL-X POIN 140
C            BOUNDARY, OR                             POIN 150
C            2. AN INTERNAL RADIAL-X FLOW COOLANT BLOCK IS NOT POIN 160
C            TRAVERSED BY AT LEAST ONE RADIAL-X GRID LINE. POIN 170
C  POINTS8   1. AN EXTERNAL COOLANT IS FLOWING INTO AN AXIAL-Z POIN 180
C            BOUNDARY, OR                             POIN 190
C            2. AN INTERNAL AXIAL-Z FLOW COOLANT BLOCK IS NOT POIN 200
C            TRAVERSED BY AT LEAST ONE AXIAL-Z GRID LINE. POIN 210
C  POINTS9   1. AN EXTERNAL COOLANT IS FLOWING INTO A THETA-Y POIN 220
C            BOUNDARY, OR                             POIN 230
C            2. AN INTERNAL THETA-Y FLOW COOLANT BLOCK IS NOT POIN 240
C            TRAVERSED BY AT LEAST ONE THETA-Y GRID LINE. POIN 250
C  POINTS10  A GAP HAS BEEN SPECIFIED ON THE HIGH RADIAL-X POIN 260
C            BOUNDARY OF A COOLANT.                   POIN 270
C  POINTS11  A GAP HAS BEEN SPECIFIED ON THE LOW RADIAL-X POIN 280
C            BOUNDARY OF A COOLANT.                   POIN 290
C  POINTS12  A GAP HAS BEEN SPECIFIED ON THE HIGH AXIAL-Z POIN 300
C            BOUNDARY OF A COOLANT.                   POIN 310
C  POINTS13  A GAP HAS BEEN SPECIFIED ON THE LOW AXIAL-Z POIN 320
C            BOUNDARY OF A COOLANT.                   POIN 330
C  POINTS14  A GAP HAS BEEN SPECIFIED ON THE HIGH THETA-Y POIN 340
C            BOUNDARY OF A COOLANT.                   POIN 350
C  POINTS15  A GAP HAS BEEN SPECIFIED ON THE LOW THETA-Y POIN 360
C            BOUNDARY OF A COOLANT.                   POIN 370
C  =====POIN 380
C  =====POIN 390
C  ERASE THE BLOCK COMPLETENESS TABLE           POIN 400
C  DO 5 I=1,IMAX                                  POIN 410
C  DO 5 J=1,JMAX                                  POIN 420
C  DO 5 K=1,KMAX                                  POIN 430
C  MT(I,J,K)=0                                    POIN 440
5 CONTINUE                                       POIN 450

```

```

C      SET THE MESH RIBS EQUAL TO 1                                POIN 460
      DO 500 L=1,JMAX                                             POIN 470
      MT(1,L,1)=1                                                POIN 480
      MT(1,MAX,L,1)=1                                           POIN 490
      MT(1,L,KMAX)=1                                           POIN 500
500    MT(1,MAX,L,KMAX)=1                                       POIN 510
      DO 525 L=1,IMAX                                           POIN 520
      MT(L,1,1)=1                                               POIN 530
      MT(L,JMAX,1)=1                                           POIN 540
      MT(L,1,KMAX)=1                                           POIN 550
525    MT(L,JMAX,KMAX)=1                                       POIN 560
      DO 550 L=1,KMAX                                           POIN 570
      MT(1,1,L)=1                                               POIN 580
      MT(1,MAX,1,L)=1                                           POIN 590
      MT(1,JMAX,L)=1                                           POIN 600
550    MT(1,MAX,JMAX,L)=1                                       POIN 610
C=====POIN 620
C      ASSIGN THE RADIAL-X BLOCKS AND FIND THE RADIAL-X GAPS    POIN 630
      CALL BOUNDA(RL,RP,IM,IGR,RBL,RBH,RDG,IL,IH,NRG,1)         POIN 640
C      ARE THERE TOO MANY RADIAL-X GAPS                          POIN 650
      IF(NRG.GT.IAXRG) CALL ERROR(7HPOINTS2)                   POIN 660
C=====POIN 670
C      ASSIGN THE AXIAL-Z BLOCKS AND FIND THE AXIAL-Z GAPS    POIN 680
      CALL BOUNDA(ZL,ZP,JM,JGZ,ZBL,ZBH,ZDG,JL,JH,NZG,2)         POIN 690
C      ARE THERE TOO MANY AXIAL-Z GAPS                          POIN 700
      IF(NZG.GT.IAXZG) CALL ERROR(7HPOINTS3)                   POIN 710
C=====POIN 720
C      ASSIGN THE THETA-Y BLOCKS AND FIND THE THETA-Y GAPS    POIN 730
      CALL BOUNDA(TL,TP,KM,KGT,TBL,TBH,TDG,KL,KH,NTG,3)         POIN 740
C      ARE THERE TOO MANY THETA-Y GAPS                          POIN 750
      IF(NTG.GT.MAXTG) CALL ERROR(7HPOINTS4)                   POIN 760
C=====POIN 770
C      INITIALIZE ARRAYS.                                       POIN 780
      DO 6 IG=1,MAXRG                                           POIN 790
      DO 6 J=1,MAXZP                                             POIN 800
      DO 6 K=1,MAXTP                                             POIN 810
      RUBIL(IG,J,K)=0.0                                         POIN 820
      RBBTH(IG,J,K)=0.0                                         POIN 830
      GAPK(IG,J,K)=0.0                                          POIN 840
      MATNG(IG,J,K)=0                                           POIN 850
6      CONTINUE                                                POIN 860
      DO 7 JG=1,MAXZG                                           POIN 870
      DO 7 I=1,MAXRP                                             POIN 880
      DO 7 K=1,MAXTP                                             POIN 890
      ZBBIL(JG,I,K)=0.0                                         POIN 900
      ZBBTH(JG,I,K)=0.0                                         POIN 910

```

```

GAPZ(JG,I,K)=0.0
MATZG(JG,I,K)=0
7 CONTINUE
DO 8 KG=1,MAXTG
DO 8 I=1,MAXRP
DO 8 J=1,MAXZP
TBBIL(KG,I,J)=0.0
TBBIH(KG,I,J)=0.0
GAPI(KG,I,J)=0.0
MATIG(KG,I,J)=0.0
8 CONTINUE
=====POIN1030
C LOOK AT EVERY BLOCK POIN1040
DO 120 L=1,LMAX POIN1050
C ASSIGN THE CORNERS OF THE CURRENT BLOCK POIN1060
ILS=IL(L) POIN1070
IHS=IH(L) POIN1080
JLS=JL(L) POIN1090
JHS=JH(L) POIN1100
KLS=KL(L) POIN1110
KHS=KH(L) POIN1120
C IS THIS BLOCK A COOLANT OR A SOLID MATERIAL? -'COOL.,SOLID'- POIN1130
IF (MB(L),LE.0) GO TO 115 POIN1140
=====POIN1150
C IS A RADIAL-X GAP PRESENT? -'NO,YES'- POIN1160
IF (RDG(L),LE.0.)GO TO 30 POIN1170
=====POIN1180
C ASSIGN THE RADIAL-X GAP POIN1190
IG=IGR(IHS) POIN1200
DO 20 J=JLS,JHS POIN1210
DO 10 K=KLS,KHS POIN1220
C HAS A COOLANT BEEN DEFINED ADJACENT TO THE GAP? POIN1230
IF (GAPR(IG,J,K) .GE. 0.0) GO TO 15 POIN1240
CALL ERROR(8HPOINTS11) POIN1250
GO TO 100 POIN1260
15 GAPR(IG,J,K)=RDG(L) POIN1270
MATIG(IG,J,K)=MGR(L) POIN1280
RBBIL(IG,J,K)=460.0 POIN1290
RBBIH(IG,J,K)=460.0 POIN1300
10 CONTINUE POIN1310
20 CONTINUE POIN1320
=====POIN1330
C IS AN AXIAL-Z GAP PRESENT? -'NO,YES'- POIN1340
30 IF (ZDG(L),LE.0,0)GO TO 60 POIN1350
=====POIN1360
C ASSIGN THE AXIAL-Z GAP POIN1370

```


	JG=JGZ(JHS)	POIN1380
	DO 50 I=ILS,IHS	POIN1390
	DO 40 K=KLS,KHS	POIN1400
C	HAS A COOLANT BEEN DEFINED ADJACENT TO THE GAP?	POIN1410
	IF (GAPZ(JG,I,K) .GE. 0.0) GO TO 45	POIN1420
	CALL ERROR(BHPOINTS13)	POIN1430
	GO TO 100	POIN1440
45	GAPZ(JG,I,K)=ZOG(L)	POIN1450
	MATZG(JG,I,K)=MGZ(L)	POIN1460
	ZBBIL(JG,I,K)=460.0	POIN1470
	ZBBIH(JG,I,K)=460.0	POIN1480
40	CONTINUE	POIN1490
50	CONTINUE	POIN1500
C=====		POIN1510
C	IS A THETA-Y GAP PRESENT? -'NO,YES'-	POIN1520
60	IF (IDG(L) .LE. 0.0) GO TO 100	POIN1530
C=====		POIN1540
C	ASSIGN THE THETA-Y GAP	POIN1550
	KG=KGT(KHS)	POIN1560
	DO 80 I=ILS,IHS	POIN1570
	DO 70 J=JLS,JHS	POIN1580
C	HAS A COOLANT BEEN DEFINED ADJACENT TO THE GAP?	POIN1590
	IF (GAPI(KG,I,J) .GE. 0.0) GO TO 75	POIN1600
	CALL ERROR (BHPOINTS15)	POIN1610
	GO TO 100	POIN1620
75	GAPI(KG,I,J)=TDG(L)	POIN1630
	MATIG(KG,I,J)=MGT(L)	POIN1640
	TBBIL(KG,I,J)=460.0	POIN1650
	TBBIH(KG,I,J)=460.0	POIN1660
70	CONTINUE	POIN1670
80	CONTINUE	POIN1680
C=====		POIN1690
C/	RECORD ALL POINTS BELONGING TO THE CURRENT BLOCK -'IMPOS,OK'-	POIN1700
100	DO 110 I=ILS,IHS	POIN1710
	DO 110 J=JLS,JHS	POIN1720
	DO 110 K=KLS,KHS	POIN1730
	IF (MT(I,J,K) .NE. 0) CALL ERROR (7HPOINTS5)	POIN1740
	MT(I,J,K)=1	POIN1750
110	CONTINUE	POIN1760
	GO TO 120	POIN1770
CE	*****	POIN1780
C=====		POIN1790
C	ASSIGN THE COOLANT NUMBER	POIN1800
115	J=IABS(MB(L))	POIN1810
C	ASSIGN THE FLOW DIRECTION	POIN1820
	IP=IABS(IPATH(J))	POIN1830

```

C      IS IT A VALID DIRECTION? -'YES,NO'-                                POIN1840
C      IF (IP.GT.0.AND.IP.LT.4) GO TO 200                                POIN1850
C      IMPOSSIBLE FLOW DIRECTION NUMBER                                POIN1860
C      CALL ERROR(7HPOINTS6)                                           POIN1870
C      IS AN EXTERNAL COOLANT FLOWING INTO A BOUNDARY ?                POIN1880
C      DOES AN INTERNAL COOLANT FLOW THROUGH ONLY ONE LEVEL OF POINTS ? POIN1890
200  IF (IHS.EQ.ILS.AND.IP.EQ.1) CALL ERROR(7HPOINTS7)                   POIN1900
C      IF (JHS.EQ.JLS.AND.IP.EQ.2) CALL ERROR(7HPOINTS8)                 POIN1910
C      IF (KHS.EQ.KLS.AND.IP.EQ.3) CALL ERROR(7HPOINTS9)                 POIN1920
C=====POIN1930
C      IS THE FLOW IN THE RADIAL-X DIRECTION? -'YES,NO'-                POIN1940
C      IF (IP.EQ.1) GO TO 220                                           POIN1950
C      SET TWO SWITCHES TO INDICATE ON WHICH SIDE OF ITS RADIAL-X      POIN1960
C      BOUNDARIES THE COOLANT LIES                                       POIN1970
C      IG1= IGR(IHS)                                                     POIN1980
C      IG2= IGR(ILS-1)                                                   POIN1990
C      IF (ILS.LE.1) IG2=0                                               POIN2000
C      DO 210J= JLS,JHS                                                 POIN2010
C      DO 210K= KLS,KHS                                                 POIN2020
C      IF (IG2.LE.0) GO TO 180                                           POIN2030
C      IS THERE A GAP ADJACENT TO COOLANT'S LOW BOUNDARY?              POIN2040
C      IF (GAPR(IG2,J,K) .LE. 0.0) GO TO 175                            POIN2050
C      CALL ERROR(8HPOINTS11)                                           POIN2060
C      GO TO 100                                                         POIN2070
175  GAPR(IG2,J,K)=-1.0E-10+GAPR(IG2,J,K)                               POIN2080
C      RBB1L(IG2,J,K)=460.0                                             POIN2090
C      RBB1H(IG2,J,K)=460.0                                             POIN2100
180  IF (IG1.LE.0) GO TO 210                                           POIN2110
C      IS THERE A GAP ADJACENT TO COOLANT HIGH BOUNDARY?               POIN2120
C      IF (GAPR(IG1,J,K) .LE. 0.0) GO TO 184                            POIN2130
C      CALL ERROR(8HPOINTS10)                                           POIN2140
C      GO TO 100                                                         POIN2150
184  GAPR(IG1,J,K)=-2.0E-10+GAPR(IG1,J,K)                               POIN2160
C      RBB1L(IG1,J,K)=460.0                                             POIN2170
C      RBB1H(IG1,J,K)=460.0                                             POIN2180
210  CONTINUE                                                           POIN2190
C=====POIN2200
C      IS THE FLOW IN THE AXIAL-Z DIRECTION? -'YES,NO'-                POIN2210
C      220 IF (IP.EQ.2) GO TO 240                                        POIN2220
C      SET TWO SWITCHES TO INDICATE ON WHICH SIDE OF ITS AXIAL-Z      POIN2230
C      BOUNDARIES THE COOLANT LIES                                       POIN2240
C      JG1= JG2(JHS)                                                     POIN2250
C      JG2= JG2(JLS-1)                                                   POIN2260
C      IF (JLS.LE.1) JG2=0                                               POIN2270
C      DO 230 I=ILS,IHS                                                 POIN2280
C      DO 230 K= KLS,KHS                                                 POIN2290

```

```

C      IF (JG1.LE.0) GO TO 215
C      IS THERE A GAP ADJACENT TO COOLANT HIGH BOUNDARY?
C      IF (GAPZ(JG1,I,K) .LE. 0.0) GO TO 213
C      CALL ERROR(BHPOINTS12)
C      GO TO 100
213  GAPZ(JG1,I,K)=-2.0E-10+GAPZ(JG1,I,K)
C      ZBBIL(JG1,I,K)=460.0
C      ZBBIH(JG1,I,K)=460.0
215  IF (JG2.LE.0) GO TO 230
C      IS THERE A GAP ADJACENT TO COOLANT'S LOW BOUNDARY?
C      IF (GAPZ(JG2,I,K) .LE. 0.0) GO TO 218
C      CALL ERROR(BHPOINTS13)
C      GO TO 100
218  GAPZ(JG2,I,K)=-1.0E-10+GAPZ(JG2,I,K)
C      ZBBIL(JG2,I,K)=460.0
C      ZBBIH(JG2,I,K)=460.0
230  CONTINUE
C=====
C      IS THE FLOW IN THE THETA-Y DIRECTION? -'YES,NO'-
240  IF (IF.LE.3) GO TO 100
C      SET TWO SWITCHES TO INDICATE ON WHICH SIDE OF ITS THETA-Y
C      BOUNDARIES THE COOLANT LIES
C      KG1= KG1(KHS)
C      KG2= KG1(KLS-1)
C      IF (KLS.LE.1) KG2=0
C      DO 250 I= ILS,IHS
C      DO 250 J= JLS,JHS
C      IF (KG1.LE.0) GO TO 255
C      IS THERE A GAP ADJACENT TO COOLANT HIGH BOUNDARY?
C      IF (GAPT(KG1,I,J) .LE. 0.0) GO TO 253
C      CALL ERROR (AHPOINTS14)
C      GO TO 100
253  GAPT(KG1,I,J)=-2.0E-10+GAPT(KG1,I,J)
C      TBBIL(KG1,I,J)=460.0
C      TBBIH(KG1,I,J)=460.0
255  IF (KG2.LE.0) GO TO 250
C      IS THERE A GAP ADJACENT TO COOLANT'S LOW BOUNDARY?
C      IF (GAPT(KG2,I,J) .LE. 0.0) GO TO 257
C      CALL ERROR (AHPOINTS15)
C      GO TO 100
257  GAPT(KG2,I,J)=-1.0E-10+GAPT(KG2,I,J)
C      TBBIL(KG2,I,J)=460.0
C      TBBIH(KG2,I,J)=460.0
250  CONTINUE
C      GO TO 100
C=====

```

```

POIN2300
POIN2310
POIN2320
POIN2330
POIN2340
POIN2350
POIN2360
POIN2370
POIN2380
POIN2390
POIN2400
POIN2410
POIN2420
POIN2430
POIN2440
POIN2450
POIN2460
POIN2470
POIN2480
POIN2490
POIN2500
POIN2510
POIN2520
POIN2530
POIN2540
POIN2550
POIN2560
POIN2570
POIN2580
POIN2590
POIN2600
POIN2610
POIN2620
POIN2630
POIN2640
POIN2650
POIN2660
POIN2670
POIN2680
POIN2690
POIN2700
POIN2710
POIN2720
POIN2730
POIN2740
POIN2750

```

```

C      HAVE ALL BLOCKS BEEN CHECKED? -'YES,NOTYET'-          POIN2760
120 CONTINUE                                                POIN2770
C=====POIN2780
C      INITIALIZE THE MATRG,MATZG AND MATTG VALUES WHICH ARE ON GAPLINES POIN2790
C      BOUNDING COOLANTS.                                     POIN2800
      DO 260 I=1,NRG                                          POIN2810
      DO 260 J=1,JMAX                                         POIN2820
      DO 260 K=1,KMAX                                         POIN2830
260 IF(GAPR(I,J,K).LT.-.5E-10) MATRG(I,J,K)=100           POIN2840
      DO 270 I=1,IMAX                                         POIN2850
      DO 270 J=1,NZG                                          POIN2860
      DO 270 K=1,KMAX                                         POIN2870
270 IF(GAPZ(I,J,K).LT.-.5E-10) MATZG(I,J,K)=100           POIN2880
      DO 280 I=1,IMAX                                         POIN2890
      DO 280 J=1,JMAX                                         POIN2900
      DO 280 K=1,NTG                                          POIN2910
280 IF(GAPT(I,J,K).LT.-.5E-10) MATTG(I,J,K)=100           POIN2920
C=====POIN2930
C/      HAVE ALL POINTS BEEN ASSIGNED TO A BLOCK? -'NO,YES'- POIN2940
      DO 150 I=1,IMAX                                         POIN2950
      DO 140 J=1,JMAX                                         POIN2960
      DO 130 K=1,KMAX                                         POIN2970
      IF(MT(I,J,K).LE.0) CALL ERROR(7HPPOINTS1)              POIN2980
130 CONTINUE                                                POIN2990
140 CONTINUE                                                POIN3000
150 CONTINUE                                                POIN3010
CE *****POIN3020
      RETURN                                                POIN3030
C=====POIN3040
      END                                                    POIN3050

```

```

SUBROUTINE BOUNDA(AL,AP,M,IGA,ABL,ABH,ADG,ILA,IHA,NAG,IP)      BOUN 10
INCLUDE COMDIM                                                BOUN 20
C=====
C ASSIGN THE BLOCK AND GAP LOCATIONS FOR ONE DIMENSION      BOUN 30
C=====
C ERROR STOPS=                                              BOUN 60
C   BOUNDA1 THE LOW BOUNDARY FOR A BLOCK IS TOO LARGE OR    BOUN 70
C           LARGER THAN THE LARGEST BOUNDARY DEFINED IN THAT BOUN 80
C           DIMENSION.                                       BOUN 90
C   BOUNDA2 THE HIGH BOUNDARY OF A BLOCK IS LARGER THAN THE BOUN 100
C           LARGEST BOUNDARY DEFINED IN THAT DIMENSION.     BOUN 110
C   BOUNDA3 AN INTERNAL COOLANT BLOCK HAS BEEN SPECIFIED IN BOUN 120
C           VIOLATION OF CODE LIMITATIONS. MOST LIKELY ERRORS ARE BOUN 130
C           1. A NORMAL COOLANT HAS BEEN SPECIFIED FOR A    BOUN 140
C           PROBLEM IN CYLINDRICAL COORDINATES.             BOUN 150
C           2. AN INTERNAL COOLANT HAS TWO BOUNDARIES WHICH BOUN 160
C           ARE COINCIDENT.                                   BOUN 170
C=====
C=====
C DIMENSION AL ( 1),AP ( 1),IGA ( 1),ABL ( 1), ABH ( 1),ADG ( 1),ILA ( 1),IHA ( 1) BOUN 200
C 1 ABH ( 1),ADG ( 1),ILA ( 1),IHA ( 1) BOUN 210
C=====
C CALCULATE THE POINT LOCATIONS                              BOUN 230
AP(1)=AL(1)                                                  BOUN 240
DO 10 I=2,M                                                  BOUN 250
AP(I)=(AL(I)+AL(I-1))*0.5                                    BOUN 260
10 CONTINUE                                                  BOUN 270
AP(M+1)=AL(M)                                               BOUN 280
C=====
C ERASE THE GAP LOCATION ARRAY                               BOUN 300
DO 20 I=1,M                                                  BOUN 310
IGA(I)=0                                                     BOUN 320
20 CONTINUE                                                  BOUN 330
M1=M+1                                                       BOUN 340
C=====
C FIND THE BLOCK BOUNDARY INDICES IN THE CURRENT DIMENSION BOUN 350
C .FOR EACH BLOCK                                           BOUN 370
DO 70 L=1,LMAX                                              BOUN 380
C IS IT A COOLANT OR A SOLID MATERIAL BLOCK? -SOLID,COOL,- BOUN 390
IF(MB(L).GT.0)GO TO 25                                       BOUN 400
C IS IT AN EXTERNAL OR INTERNAL COOLANT? -INTERN,EXTERN,- BOUN 410
IF(ABL(L).NE.ABH(L))GO TO 25                                   BOUN 420
CB ASSIGN THE EXTERNAL COOLANT BLOCK                          BOUN 430
C IS IT ON THE LOW INDEX OUTSIDE -YES,NO,-                  BOUN 440
IF(ABL(L).EQ.AP(1))GO TO 105                                  BOUN 450

```

```

C      IS IT ON THE HIGH INDEX OUTSIDE -YES,NO:- BOUN 460
      IF(ABS(L).EQ.AP(M1))GO TO 110 BOUN 470
C      ILLEGAL INTERNAL COOLANT CHANNEL BOUN 480
      CALL ERROR (7HBOUND3) BOUN 490
C      ASSIGN THE LOW INDEX OUTSIDE COOLANT LIMITS BOUN 500
105  ILA(L)=1 BOUN 510
      IHA(L)=1 BOUN 520
      IGA(1)=-1 BOUN 530
      GO TO 70 BOUN 540
C      ASSIGN THE HIGH INDEX OUTSIDE COOLANT LIMITS BOUN 550
110  ILA(L)=M1 BOUN 560
      IHA(L)=M1 BOUN 570
      IGA(M)=-1 BOUN 580
      GO TO 70 BOUN 590
CE ***** BOUN 600
CB ASSIGN THE SOLID MATERIAL OR INTERNAL COOLANT BLOCK BOUN 610
C/ FIND THE LOW BLOCK BOUNDARY INDEX -OK,NONE:- BOUN 620
25  DO 30 I=2,M BOUN 630
      IF(ABS(L).LT.AP(I))GO TO 40 BOUN 640
30  CONTINUE BOUN 650
CE ***** BOUN 660
C      THE LOW BLOCK BOUNDARY LIES OUTSIDE THE SYSTEM BOUN 670
      CALL ERROR(7HBOUND1) BOUN 680
C===== BOUN 690
C      ASSIGN THE LOW BLOCK BOUNDARY BOUN 700
40  ILA(L)=1 BOUN 710
C/ FIND THE HIGH BLOCK BOUNDARY INDEX -OK,NONE:- BOUN 720
      I=I+1 BOUN 730
      DO 50 J=I,M1 BOUN 740
      IF(ABS(L).LE.AP(J))GO TO 60 BOUN 750
50  CONTINUE BOUN 760
CE ***** BOUN 770
C      THE HIGH BLOCK BOUNDARY LIES OUTSIDE THE SYSTEM BOUN 780
      CALL ERROR(7HBOUND2) BOUN 790
C      ASSIGN THE HIGH BLOCK BOUNDARY INDEX BOUN 800
60  IHA(L)=J-1 BOUN 810
C===== BOUN 820
C      SET THE GAP SWITCH WHEN NECESSARY BOUN 830
      IF(ABS(L).GT.0.0)IGA(J-1)=-1 BOUN 840
C      IS AT A COOLANT BLOCK -NO,YES:- BOUN 850
      IF(ABS(L).GT.0)GO TO 70 BOUN 860
      N=-ABS(L) BOUN 870
C      DOES THE COOLANT FLOW IN THE CURRENT DIRECTION -YES,NO:- BOUN 880
      IF (IP.LJ.IABS(IPATH(N))) GO TO 70 BOUN 890
C      ASSIGN A GAP BOUNDARY ON BOTH SIDES OF THE COOLANT BOUN 900
      IGA(J-1)=-1 BOUN 910

```

```

      IGA(I-2)=-1                                BOUN 920
C     HAVE ALL BLOCKS BEEN CHECKED? -YES,NO-     BOUN 930
      70 CONTINUE                                BOUN 940
C=====
C/    INSPECT THE GAP INDICES AND ASSING THE NUMBER OF BOUNDARIES BOUN 950
      J=0                                         BOUN 960
      DO 80 I=1,M                                 BOUN 970
      IF(IGA(I).EQ.0)GO TO 80                     BOUN 980
      J=J+1                                       BOUN 990
      IGA(I)=J                                    BOUN 1000
      80 CONTINUE                                BOUN 1010
      NAGE=J                                      BOUN 1020
CE     *****BOUN 1030
CE     *****BOUN 1040
      RETURN                                     BOUN 1050
C=====
      END                                         BOUN 1060
      BOUN 1070
      BOUN 1080

```

```

SUBROUTINE CONSTA                                CONS 10
INCLUDE      COMDIM                              CONS 20
C=====CONS 30
C/  CALCULATE RADIAL-X GEOMETRIC CORRECTION FACTOR 1 (RLN TABLE C1)  CONS 40
C  =====CONS 50
C  =====CONS 60
      DO 10 I=1,IM                                CONS 70
      DELR(I)=RP(I+1)-RP(I)                        CONS 80
      IF (ISHAPE.EQ.1) GO TO 5                      CONS 90
      RLN(I)=ALOG(RL(I)/RP(I))/ALOG(RP(I+1)/RP(I))  CONS 100
      GO TO 10
      5  RLN(I)=(RL(I)-RP(I))/DELR(I)              CONS 110
      10 CONTINUE                                  CONS 120
C  CALCULATE AXIAL-Z GEOMETRIC CORRECTION FACTOR 1 (ZLN TABLE C1)  CONS 140
      DO 20 J=1,JM                                CONS 150
      DELZ(J)=ZP(J+1)-ZP(J)                      CONS 160
      ZLN(J)=(ZL(J)-ZP(J))/DELZ(J)               CONS 170
      20 CONTINUE                                  CONS 180
C  CALCULATE THETA-Y GEOMETRIC CORRECTION FACTOR 1 (TLN TABLE C1)  CONS 190
      DO 30 K=1,KM                                CONS 200
      DELT(K)=TP(K+1)-TP(K)                      CONS 210
      TLN(K)=(TL(K)-TP(K))/DELT(K)               CONS 220
      30 CONTINUE                                  CONS 230
CE *****CONS 240
C/  CALCULATE RADIAL-X GEOMETRIC GAP CORRECTION FACTORS 2 AND 3    CONS 250
C  (RATIOK,RATIOH, TABLES C2,C4)                                CONS 260
      DO 40 I=1,IM1                                CONS 270
      IF (IGK(I).LE.0) GO TO 40                    CONS 280
      IG=IGK(I)                                    CONS 290
      IF (ISHAPE.EQ.1) GO TO 35                    CONS 300
      RATIOK(IG)=ALOG(RP(I+1)/RP(I))/ALOG(RP(I+1)/RL(I))  CONS 310
      RATIOH(IG)=RL(I)*ALOG(RP(I+1)/RL(I))          CONS 320
      RATIOG(IG)=1.0/(RL(I)*ALOG(RP(I+1)/RP(I)))    CONS 330
      GO TO 40
      35 RATIOK(IG)=DELR(I)/(RP(I+1)-RL(I))          CONS 350
      RATIOH(IG)=RP(I+1)-RL(I)                    CONS 360
      RATIOG(IG)=1.0/DELR(I)                      CONS 370
      40 CONTINUE                                  CONS 380
      RATIOK(NRG)=0.0                              CONS 390
      RATIOH(NRG)=0.0                              CONS 400
      IF (ISHAPE.EQ.1) GO TO 43                    CONS 410
      RATIOG(NRG)=1.0/(RL(IM)*ALOG(RP(IM+1)/RP(IM)))  CONS 420
      GO TO 45
      43 RATIOG(NRG)=1.0/DELR(IM)                  CONS 440
      45 CONTINUE                                  CONS 450

```



```

CE *****CONS 460
C/ CALCULATE AXIAL-Z GEOMETRIC GAP CORRECTION FACTORS 2 AND 3 CONS 470
C .(ZAT1OK,ZATIOH, TABLES C2,C4) CONS 480
DO 50 J=1,JM1 CONS 490
IF(JGZ(J).LE.0)GO TO 50 CONS 500
JG=JGZ(J) CONS 510
ZAT1OK(JG)=DELZ(J)/(ZP(J+1)-ZL(J)) CONS 520
ZATIOH(JG)=ZP(J+1)-ZL(J) CONS 530
50 CONTINUE CONS 540
ZAT1OK(NZG)=0.0 CONS 550
ZATIOH(NZG)=0.0 CONS 560
CE *****CONS 570
C/ CALCULATE THE TA-Y GEOMETRIC GAP CORRECTION FACTORS 2 AND 3 CONS 580
C .(TAT1OK,TATIOH, TABLES C2,C4) CONS 590
DO 60 K=1,KM1 CONS 600
IF(KGT(K).LE.0)GO TO 60 CONS 610
KG=KGT(K) CONS 620
TAT1OK(KG)=DELT(K)/(TP(K+1)-TL(K)) CONS 630
TATIOH(KG)=TP(K+1)-TL(K) CONS 640
60 CONTINUE CONS 650
TAT1OK(NTG)=0.0 CONS 660
TATIOH(NTG)=0.0 CONS 670
CE *****CONS 680
CB PASS THROUGH ALL BLOCKS TO CALCULATE THE GAP CORRECTION FACTORS CONS 690
C DEFINE THE CORNERS OF THE CURRENT BLOCK CONS 710
ILS= IL(L) CONS 720
IHS= IH(L) CONS 730
JLS= JL(L) CONS 740
JHS= JH(L) CONS 750
KLS= KL(L) CONS 760
KHS= KH(L) CONS 770
C IS IT A COOLANT OR A SOLID MATERIAL BLOCK? -,'COOL.',SOLID'- CONS 780
IF(MB(L).LE.0)GO TO 110 CONS 790
C=====CONS 800
C/ CALCULATE RADIAL-X GEOMETRIC GAP CORRECTION FACTOR 4 CONS 810
C .(RAT1OB, TABLE C3) CONS 820
IF(RDG(L).LE.0.0)GO TO 90 CONS 830
IG=IGK(IHS) CONS 840
IF(ISHAPE.EQ.1)GO TO 70 CONS 850
X=ALOG((RL(IHS)-RDG(L))/RP(IHS))/ALOG(RP(IHS+1)/RL(IHS)) CONS 860
GO TO 75 CONS 870
70 X=(RL(IHS)-RP(IHS)-RDG(L))/(RP(IHS+1)-RL(IHS)) CONS 880
75 DO 80 J=JLS,JHS CONS 890
DO 80 K=KLS,KHS CONS 900
RAT1OB(IG,J,K)=X CONS 910

```

```

CE *****CONS 460
C/ CALCULATE AXIAL-Z GEOMETRIC GAP CORRECTION FACTORS 2 AND 3 CONS 470
C .(ZATIOK,ZATIOH, TABLES C2,C4) CONS 480
DO 50 J=1,JM1 CONS 490
IF(JGZ(J).LE.0)GO TO 50 CONS 500
JG=JGZ(J) CONS 510
ZATIOK(JG)=DELZ(J)/(ZP(J+1)-ZL(J)) CONS 520
ZATIOH(JG)=ZP(J+1)-ZL(J) CONS 530
50 CONTINUE CONS 540
ZATIOK(NZG)=0.0 CONS 550
ZATIOH(NZG)=0.0 CONS 560
CE *****CONS 570
C/ CALCULATE THETA-Y GEOMETRIC GAP CORRECTION FACTORS 2 AND 3 CONS 580
C .(TATIOK,TATIOH, TABLES C2,C4) CONS 590
DO 60 K=1,KM1 CONS 600
IF(KGT(K).LE.0)GO TO 60 CONS 610
KG=KGT(K) CONS 620
TATIOK(KG)=DELT(K)/(TP(K+1)-TL(K)) CONS 630
TATIOH(KG)=TP(K+1)-TL(K) CONS 640
60 CONTINUE CONS 650
TATIOK(NTG)=0.0 CONS 660
TATIOH(NTG)=0.0 CONS 670
CE *****CONS 680
CB PASS THROUGH ALL BLOCKS TO CALCULATE THE GAP CORRECTION FACTORS CONS 690
DO 120 L=1,LMAX CONS 700
C DEFINE THE CORNERS OF THE CURRENT BLOCK CONS 710
IL= IL(L) CONS 720
IH= IH(L) CONS 730
JL= JL(L) CONS 740
JH= JH(L) CONS 750
KL= KL(L) CONS 760
KH= KH(L) CONS 770
C IS IT A COOLANT OR A SOLID MATERIAL BLOCK? --COOL.,SOLID-- CONS 780
IF(IH(L).LE.0)GO TO 110 CONS 790
C=====CONS 800
C/ CALCULATE RADIAL-X GEOMETRIC GAP CORRECTION FACTOR 4 CONS 810
C .(RATIOB, TABLE C3) CONS 820
IF(RDG(L).LE.0.0)GO TO 90 CONS 830
IG=IGK(IHS) CONS 840
IF(ISHAPE.EQ.1)GO TO 70 CONS 850
X=ALOG((RL(IHS)-RDG(L))/RP(IHS))/ALOG(RP(IHS+1)/RL(IHS)) CONS 860
GO TO 75 CONS 870
70 X=(KL(IHS)-RP(IHS)-RDG(L))/(RP(IHS+1)-RL(IHS)) CONS 880
75 DO 80 J=JL,JHS CONS 890
DO 80 K=KL,KHS CONS 900
RATIOB(IG,J,K)=X CONS 910

```

```

80 CONTINUE
=====
C CALCULATE AXIAL-Z GEOMETRIC GAP CORRECTION FACTOR 4
C .(ZAT10B, TABLE C3)
90 IF(ZD6(L).LE.0.0)GO TO 100
   JG=JGZ(JHS)
   X=(ZL(JHS)-ZP(JHS)-ZD6(L))/(ZP(JHS+1)-ZL(JHS))
   DO 95 I=ILS,IHS
   DO 95 K=KLS,KHS
   ZAT10B(JG,I,K)=X
95 CONTINUE
=====
C CALCULATE THETA-Y GEOMETRIC GAP CORRECTION FACTOR 4
C .(TAT10B, TABLE C3)
100 IF(TD6(L).LE.0.0)GO TO 120
   KG=KGT(KHS)
   X=(TL(KHS)-TP(KHS)-TD6(L))/(TP(KHS+1)-TL(KHS))
   DO 105 I=ILS,IHS
   DO 105 J=JLS,JHS
   TAT10B(KG,I,J)=X
105 CONTINUE
   GO TO 120
CE *****
C/ CALCULATE THE RADIAL-X CORRECTION FOR RADIATION ACROSS A COOLANT
C AVOID OUTSIDE RADIAL-X COOLANTS
110 IF (ILS.EQ. 1 .OR. IHS .EQ. IMAX) GO TO 212
   IF(ISHAPE.EQ.1) GO TO 214
   X=RL(ILS-1)/RL(IHS)
   GO TO 216
212 X=0.0
   GO TO 216
214 X=1.0
216 IG=IGK(IHS)
   DO 115 J=JLS,JHS
   DO 115 K=KLS,KHS
   RAT10B(IG,J,K)=X
115 CONTINUE
CE *****
C HAVE ALL BLOCKS BEEN CHECKED? -'YES,NO'-
120 CONTINUE
   RETURN
   END
=====
CONS 920
CONS 930
CONS 940
CONS 950
CONS 960
CONS 970
CONS 980
CONS 990
CONS1000
CONS1010
CONS1020
CONS1030
CONS1040
CONS1050
CONS1060
CONS1070
CONS1080
CONS1090
CONS1100
CONS1110
CONS1120
CONS1130
CONS1140
CONS1150
CONS1160
CONS1170
CONS1180
CONS1190
CONS1200
CONS1210
CONS1220
CONS1230
CONS1240
CONS1250
CONS1260
CONS1270
CONS1280
CONS1290
CONS1300
CONS1310
CONS1320
CONS1330

```

```

SUBROUTINE GEOMET                                GEOM 10
INCLUDE COMDIM                                    GEOM 20
C=====GEOM 30
CB CALCULATE THE GEOMETRICAL FACTOR AND CELL VOLUMES FOR NODAL POINTS GEOM 40
C=====GEOM 50
C ERROR STOPS                                     GEOM 60
C GEOMLT1 SOME POINT HAS A NEGATIVE CALCULATED VOLUME GEOM 70
C=====GEOM 80
C=====GEOM 90
LOGICAL IGAP JGAP KGAP                           GEOM 100
C=====GEOM 110
C PASS THROUGH ALL PLANES IN THE RADIAL-X DIRECTION GEOM 120
DO 400 I=1,IM                                     GEOM 130
C IS A RADIAL-X GAP ADJACENT? -'NO,YES'-          GEOM 140
IF (IGR(I).EQ.0) GO TO 10                          GEOM 150
C=====GEOM 160
C ASSIGN THE RADIAL-X GAP INDEX                    GEOM 170
IG=IGR(I)                                         GEOM 180
IGAP=.TRUE.                                       GEOM 190
GO TO 20                                          GEOM 200
C=====GEOM 210
C SET THE RADIAL-X GAP AVERAGES TO ZERO           GEOM 220
10 IGAP=.FALSE.                                   GEOM 230
JGJK=0.0                                         GEOM 240
JGKJ=0.0                                         GEOM 250
JGR=0.0                                          GEOM 260
C=====GEOM 270
C FOR A GIVEN RADIAL-X PLANE, PASS THROUGH THE AXIAL-Z COLUMNS GEOM 280
20 DO 300 J=1,JM                                  GEOM 290
C IS AN AXIAL-Z GAP ADJACENT? -'NO,YES'-          GEOM 300
IF (JGZ(J).EQ.0) GO TO 30                         GEOM 310
C=====GEOM 320
C ASSIGN THE AXIAL-Z GAP INDEX                    GEOM 330
JG=JGZ(J)                                        GEOM 340
JGAP=.TRUE.                                       GEOM 350
GO TO 40                                          GEOM 360
C=====GEOM 370
C SET THE AXIAL-Z GAP AVERAGES TO ZERO           GEOM 380
30 JGAP=.FALSE.                                   GEOM 390
JGJK=0.0                                         GEOM 400
JGKJ=0.0                                         GEOM 410
JGZ=0.0                                          GEOM 420
C=====GEOM 430
C FOR THE GIVEN AXIAL-Z COLUMNS, PASS THROUGH ALL THETA-Y POINTS GEOM 440
40 DO 400 K=1,KM                                  GEOM 450

```

```

C      IS A THETA-Y GAP ADJACENT? -'NO,YES'-          GEOM 460
      IF (KGT(K).EQ.0) GO TO 50                        GEOM 470
C=====
C      ASSIGN THE THETA-Y GAP INDEX                    GEOM 480
      KG= KGT(K)                                       GEOM 490
      KGAP= .TRUE.                                     GEOM 500
      GO TO 70                                         GEOM 510
C=====
C      SET THE THETA-Y GAP AVERAGES TO ZERO           GEOM 520
      50 KGAP= .FALSE.                                  GEOM 530
      DGJ= 0.0                                         GEOM 540
      DGJ1= 0.0                                        GEOM 550
      DGT= 0.0                                         GEOM 560
C=====
C      IS A RADIAL-X GAP ADJACENT? -'NO,YES'-        GEOM 570
      70 IF (.NOT.IGAP) GO TO 80                       GEOM 580
C=====
C      ASSIGN THE RADIAL-X GAP AVERAGES              GEOM 590
      DGJK= (GAPR(IG,J,K)+GAPR(IG,J+1,K))*0.5        GEOM 600
      DGKJ= (GAPR(IG,J,K)+GAPR(IG,J,K+1))*0.5        GEOM 610
      DGR= GAPR(IG,J,K)                                GEOM 620
C=====
C      IS AN AXIAL-Z GAP ADJACENT? -'NO,YES'-        GEOM 630
      80 IF (.NOT.JGAP) GO TO 90                       GEOM 640
C=====
C      ASSIGN THE AXIAL-Z GAP AVERAGES               GEOM 650
      DGIZ= (GAPZ(JG,I,K)+GAPZ(JG,I+1,K))*0.5        GEOM 660
      DGKI= (GAPZ(JG,I,K)+GAPZ(JG,I,K+1))*0.5        GEOM 670
      DJZ= GAPZ(JG,I,K)                                GEOM 680
C=====
C      IS A THETA-Y GAP ADJACENT? -'NO,YES'-        GEOM 690
      90 IF (.NOT.KGAP) GO TO 100                      GEOM 700
C=====
C      ASSIGN THE THETA-Y GAP ZVERAGES              GEOM 710
      DGJ= (GAPT(KG,I,J)+GAPT(KG,I+1,J))*0.5          GEOM 720
      DGJ1= (GAPT(KG,I,J)+GAPT(KG,I,J+1))*0.5         GEOM 730
      DGT= GAPT(KG,I,J)                                GEOM 740
C=====
C      IS THE GEOMETRY CYLINDRICAL OR RECTANGULAR? -'RECT.,CYLIN.'- GEOM 750
      100 RLX=RL(I-1)                                  GEOM 760
      IF (I.EQ.1) RLX=0.0                              GEOM 770
      ZLX=ZL(J-1)                                      GEOM 780
      IF (J.EQ.1) ZLX=0.0                              GEOM 790
      TLX=TL(K-1)                                      GEOM 800
      IF (K.EQ.1) TLX=0.0                              GEOM 810
      IF (ISHAPE.NE.0) GO TO 150                       GEOM 820
      GEOM 830
      GEOM 840
      GEOM 850
      GEOM 860
      GEOM 870
      GEOM 880
      GEOM 890
      GEOM 900
      GEOM 910

```

```

C=====GEOM 920
C   CALCULATE THE GEOMETRY FACTOR AND NODE VOLUME FOR A      GEOM 930
C   .CYLINDRICAL GEOMETRY                                     GEOM 940
      RR(1,J,K)=(ZL(J)-ZLX-DGJK)*(TL(K)-TLX-DGIJ)/ALOG(RP(I+1)/RP(I)) GEOM 950
      RZ(1,J,K)=((RL(I)-DGJK)**2-RLX**2)*(TL(K)-TLX-DGJI)/(ZP(J+1)-ZP(J) GEOM 960
X)*0.5                                                         GEOM 970
      RT(1,J,K)=(RL(I)-RLX-DGKJ)*(ZL(J)-ZLX-DGKI)/(RP(I)*(TP(K+1)-TP(K)) GEOM 980
X)                                                             GEOM 990
      V(I,J,K)=((RL(I)-DGR)**2-RLX**2)*(ZL(J)-ZLX-DGZ)*(TL(K)-TLX-DGT) GEOM,000
X*0.5                                                         GEOM1010
      GO TO 199                                               GEOM1020
C   CALCULATE THE GEOMETRY FACTOR AND NODE VOLUME FOR A      GEOM1030
C   .RECTANGULAR GEOMETRY                                     GEOM1040
150  RR(1,J,K)=(TL(K)-TLX-DGIJ)*(ZL(J)-ZLX-DGJK)/DEL R(I)    GEOM1050
      RZ(1,J,K)=(TL(K)-TLX-DGJI)*(RL(I)-RLX-DGJK)/DEL Z(J)  GEOM1060
      RT(1,J,K)=(RL(I)-RLX-DGKJ)*(ZL(J)-ZLX-DGKI)/DEL T(K)  GEOM1070
      V(I,J,K)=(TL(K)-TLX-DGT)*(RL(I)-RLX-DGR)*(ZL(J)-ZLX-DGZ) GEOM1080
199  IF (V(I,J,K) .LT. 0.0) CALL ERROR (7HGEOMET1)           GEOM1090
200  CONTINUE                                               GEOM1100
CE *****GEOM1110
300  CONTINUE                                               GEOM1120
400  CONTINUE                                               GEOM1130
C=====GEOM1140
C   CLEAN-UP CALCULATED ARRAYS SO THAT ONLY PHYSICALLY MEANINGFUL GEOM1150
C   DATA IS PRESENTED                                       GEOM1160
      DO 1000 I=1,IMAX                                       GEOM1170
      DO 1000 K=1,KMAX                                       GEOM1180
      RZ(1,JMAX,K)=0.0                                       GEOM1190
      RR(1,1,K)=0.0                                          GEOM1200
      RR(1,JMAX,K)=0.0                                       GEOM1210
      RT(1,1,K)=0.0                                          GEOM1220
      RT(1,JMAX,K)=0.0                                       GEOM1230
      V(1,1,K)=0.0                                           GEOM1240
      V(1,JMAX,K)=0.0                                       GEOM1250
1000 CONTINUE                                               GEOM1260
      DO 1500 J=1,JMAX                                       GEOM1270
      DO 1500 K=1,KMAX                                       GEOM1280
      RZ(IMAX,J,K)=0.0                                       GEOM1290
      RZ(1,J,K)=0.0                                          GEOM1300
      RT(IMAX,J,K)=0.0                                       GEOM1310
      RT(1,J,K)=0.0                                          GEOM1320
      RR(IMAX,J,K)=0.0                                       GEOM1330
      V(1,J,K)=0.0                                           GEOM1340
      V(IMAX,J,K)=0.0                                       GEOM1350
1500 CONTINUE                                               GEOM1360
      DO 2000 I=1,IMAX                                       GEOM1370

```

```

DO 2000 J=1,JMAX
RT(I,J,KMAX)=0.0
RR(I,J,1)=0.0
RR(I,J,KMAX)=0.0
RZ(I,J,1)=0.0
RZ(I,J,KMAX)=0.0
V(I,J,1)=0.0
V(I,J,KMAX)=0.0
2000 CONTINUE
C=====
C SHOULD THE NODE VOLUMES AND GEOMETRY FACTORS BE PRINTED? --NO,YES
IF(.NOT.SW(4)) RETURN
C=====
WRITE (6,103) (ZA(I), I=1,12)
WRITE (6,104)
104 FORMAT (47X,26HNODE VOLUMES (FT**3)/ )
C PRINT THE VOLUMES
CALL ARRAY1 (IDUM,V,2)
C=====
WRITE (6,103) (ZA(I), I=1,12)
WRITE (6,101) PNAME(1), PNAME(1)
101 FORMAT (15X,A6,35H GEOMETRY FACTOR (HEAT FLOW AREA/A6,52H DISTANCE
-CE) BETWEEN POINTS (I,J,K) AND (I+1,J,K),5X,4H(FT)/)
C PRINT THE RADIAL-X GEOMETRICAL FACTORS
CALL ARRAY1 (IDUM,RR,2)
C=====
WRITE (6,103) (ZA(I), I=1,12)
WRITE (6,102) PNAME(2), PNAME(2)
102 FORMAT (15X,A6,35H GEOMETRY FACTOR (HEAT FLOW AREA/A6,52H DISTANCE
-CE) BETWEEN POINTS (I,J,K) AND (I,J+1,K),5X,4H(FT)/)
C PRINT THE AXIAL-Z GEOMETRICAL FACTORS
CALL ARRAY1 (IDUM,RZ,2)
C=====
C PRINT THE THETA-Y GEOMETRICAL FACTORS
WRITE (6,103) (ZA(I), I=1,12)
WRITE (6,107) PNAME(3), PNAME(3)
107 FORMAT (15X,A6,35H GEOMETRY FACTOR (HEAT FLOW AREA/A6,52H DISTANCE
-CE) BETWEEN POINTS (I,J,K) AND (I,J,K+1),5X,4H(FT)/)
CALL ARRAY1 (IDUM,RT,2)
C=====
RETURN
C=====
103 FORMAT (1H1,30X,12A6,/)
C=====
END

```

	SUBROUTINE ARRAY1 (IX,X,IFY)	ARRA 10
	INCLUDE COMDIM	ARRA 20
C=====		ARRA 30
CB	PRINT A THREE-DIMENSIONAL ARRAY	ARRA 40
C=====		ARRA 50
C=====		ARRA 60
C=====		ARRA 70
	DIMENSION NUMBER(50),X (IQ,JQ,KQ),IX (IQ,JQ,KQ)	ARRA 80
	DATA NUMBER /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,	ARRA 90
	= 20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,	ARRA 100
	= ,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50/	ARRA 110
C=====		ARRA 120
C	WRITE THE PROPER HEADING	ARRA 130
	NADD=10	ARRA 140
	IF (IFY.EQ.1) NADD=25	ARRA 150
	GO TO (10,20,30), NPRINT	ARRA 160
10	WRITE (6,1) PNAME(3), PNAME(2)	ARRA 170
	1 FORMAT (1H0,42X,4HTHE ,A6,27H(K) DIRECTION IS HORIZONTAL/43X,	ARRA 180
	-4HTHE ,A6,25H(J) DIRECTION IS VERTICAL)	ARRA 190
	M=IMAX	ARRA 200
	N=KMAX	ARRA 210
	GO TO 40	ARRA 220
20	WRITE (6,2) PNAME(1), PNAME(3)	ARRA 230
	2 FORMAT (1H0,42X,4HTHE ,A6,27H(I) DIRECTION IS HORIZONTAL/43X,	ARRA 240
	-4HTHE ,A6,25H(K) DIRECTION IS VERTICAL)	ARRA 250
	M=JMAX	ARRA 260
	N=IMAX	ARRA 270
	GO TO 40	ARRA 280
30	WRITE (6,3) PNAME(1), PNAME(2)	ARRA 290
	3 FORMAT (1H0,42X,4HTHE ,A6,27H(I) DIRECTION IS HORIZONTAL/43X,	ARRA 300
	-4HTHE ,A6,25H(J) DIRECTION IS VERTICAL)	ARRA 310
	M=KMAX	ARRA 320
	N=IMAX	ARRA 330
40	DO 600 M=1,M1	ARRA 340
	NUM2=0	ARRA 350
100	NUM1=NUM2+1	ARRA 360
	NUM2=NUM2+NADD	ARRA 370
	IF (NUM2 .GT. N) NUM2=N	ARRA 380
	IF (NUM1 .EQ. 1) WRITE (6,4) PNAME(NPRINT), M	ARRA 390
	4 FORMAT(1H0,/,48X,A6,2H (,I2,7H) PLANE)	ARRA 400
	IF (IFY .EQ. 1) GO TO 219	ARRA 410
	WRITE(6,220) (NUMBER(L),L=NUM1,NUM2)	ARRA 420
220	FORMAT(1H0,I11,9I12)	ARRA 430
	GO TO 221	ARRA 440
219	WRITE (6,225) (NUMBER(L),L=NUM1,NUM2)	ARRA 450
225	FORMAT (1H0,5X,25I5)	


```

221 WRITE (6,170)                                ARRA 460
170 FORMAT (1H0)                                  ARRA 470
      GO TO (200,300,400),NPRINT                   ARRA 480
C=====ARRA 490
C   OUTPUT PRINT IN RADIAL-X DIRECTION, THETA-Y DIRECTION HORIZONTAL, ARRA 500
C   AXIAL-Z DIRECTION VERTICAL                     ARRA 510
C=====ARRA 520
200 DO 420 J=1,JMAX                                ARRA 530
      GO TO (421,422,423),IFY                       ARRA 540
421 WRITE (6,203) J,(IX(M,J,K),K=NUM1,NUM2)       ARRA 550
203 FORMAT (14,1X,I6,24I5)                         ARRA 560
      GO TO 420                                     ARRA 570
422 WRITE (6,201) J,(X(M,J,K),K=NUM1,NUM2)        ARRA 580
201 FORMAT (14,1P10E12.4)                          ARRA 590
      GO TO 420                                     ARRA 600
423 WRITE (6,202) J,(X(M,J,K),K=NUM1,NUM2)        ARRA 610
202 FORMAT (14,1P10E12.5)                          ARRA 620
420 CONTINUE                                       ARRA 630
      IF(NUM2 .LT. N) GO TO 100                    ARRA 640
      GO TO 600                                     ARRA 650
C=====ARRA 660
C   OUTPUT PRINT IN AXIAL-Z DIRECTION, RADIAL-X DIRECTION HORIZONTAL, ARRA 670
C   THETA-Y DIRECTION VERTICAL                     ARRA 680
C=====ARRA 690
300 DO 320 K=1,KMAX                                ARRA 700
      GO TO (321,322,323),IFY                       ARRA 710
321 WRITE (6,203) K,(IX(I,M,K), I=NUM1,NUM2)       ARRA 720
      GO TO 320                                     ARRA 730
322 WRITE (6,201) K,(X(I,M,K), I=NUM1,NUM2)        ARRA 740
      GO TO 320                                     ARRA 750
323 WRITE (6,202) K,(X(I,M,K), I=NUM1,NUM2)        ARRA 760
320 CONTINUE                                       ARRA 770
      IF(NUM2 .LT. N) GO TO 100                    ARRA 780
      GO TO 600                                     ARRA 790
C=====ARRA 800
C   OUTPUT PRINT IN THETA-Y DIRECTION, RADIAL-X DIRECTION HORIZONTAL, ARRA 810
C   AXIAL-Z DIRECTION VERTICAL                     ARRA 820
C=====ARRA 830
400 DO 230 J=1,JMAX                                ARRA 840
      GO TO (231,232,233),IFY                       ARRA 850
231 WRITE (6,203) J,(IX(I,J,M),I=NUM1,NUM2)       ARRA 860
      GO TO 230                                     ARRA 870
232 WRITE (6,201) J,( X(I,J,M),I=NUM1,NUM2)        ARRA 880
      GO TO 230                                     ARRA 890
233 WRITE (6,202) J,( X(I,J,M),I=NUM1,NUM2)        ARRA 900
230 CONTINUE                                       ARRA 910
      IF(NUM2 .LT. N) GO TO 100                    ARRA 920
600 CONTINUE                                       ARRA 930
      RETURN                                       ARRA 940
C=====ARRA 950
      END                                         ARRA 960

```

```

SUBROUTINE DPRINT
INCLUDE      COMDIM
C=====DPRI 10
CB PRINT THE POINT AND BOUNDARY ASSIGNMENTS DPRI 20
C=====DPRI 30
C/ PRINT THE RADIAL-X POINT, GRIDLINE, AND BOUNDARY ASSIGNMENTS DPRI 40
WRITE (6,100) (ZA(I), I=1,12), PNAME(1), PNAME(6), PNAME(7), DPRI 50
-PNAME(6), PNAME(7) DPRI 60
100 FORMAT(1H1,30X,12A6,///,24X,A6,23H BOUNDARY ASSIGNMENTS,///,3X, DPRI 70
-MSSEQUENCE,9X,5HPOINT,9X,9HGRID LINE,6X,14HCOOLANT OR GAP/4X, DPRI 80
-CHNUMBER,8X,CHLOCATION,8X,CHLOCATION,7X,15HBOUNDARY NUMBER/ DPRI 90
-18X,2A6,4X,2A6/) DPRI 100
DO 20 I=1,IMAX DPRI 110
TEM=RP(I)*12.0 DPRI 120
WRITE (6,107) I,TEM DPRI 130
IF (I.LE.IMAX) GO TO 20 DPRI 140
TEM=RL(I)*12.0 DPRI 150
IF (IGR(I).GT.0) GO TO 10 DPRI 160
WRITE (6,106) I,TEM DPRI 170
GO TO 20 DPRI 180
10 IG=ICK(I) DPRI 190
WRITE (6,105) I,TEM,IG DPRI 200
20 CONTINUE DPRI 210
CE *****DPRI 220
C PRINT THE AXIAL-Z POINT, GRID LINES AND BOUNDARY ASSIGNMENTS DPRI 230
WRITE (6,100) (ZA(I), I=1,12), PNAME(2), PNAME(6), PNAME(7), DPRI 240
-PNAME(6), PNAME(7) DPRI 250
DO 40 J= 1,JMAX DPRI 260
TEM= ZP(J)*12.0 DPRI 270
WRITE (6,107) J,TEM DPRI 280
IF (J.EQ.JMAX) GO TO 40 DPRI 290
TEM= ZL(J)*12.0 DPRI 300
IF (JGZ(J).GT.0) GO TO 30 DPRI 310
WRITE (6,106) J,TEM DPRI 320
GO TO 40 DPRI 330
30 JG= JGZ(J) DPRI 340
WRITE (6,105) J,TEM,JG DPRI 350
40 CONTINUE DPRI 360
CE *****DPRI 370
C/ PRINT THE THE TA-Y POINT AND BOUNDARY ASSIGNMENTS DPRI 380
WRITE (6,100) (ZA(I), I=1,12), PNAME(3), PNAME(4), PNAME(5), PNAME(4), DPRI 390
-PNAME(5) DPRI 400
DO 60 K=1,KMAX DPRI 410
TEM= TP(K)*SCALE DPRI 420
DPRI 430
DPRI 440
DPRI 450

```

```

WRITE (6,107) K,TEM                                DPRI 460
IF (K.EQ.KMAX) GO TO 60                            DPRI 470
TEM= TL(K)*SCALE                                    DPRI 480
IF (KGT(K).GT.0) GO TO 50                           DPRI 490
WRITE (6,106) K,TEM                                  DPRI 500
GO TO 60                                             DPRI 510
50 KG= KGT(K)                                        DPRI 520
WRITE (6,105)K,TEM,KG                                DPRI 530
60 CONTINUE                                          DPRI 540
CE *****DPRI 550
RETURN                                              DPRI 560
C=====DPRI 570
105 FORMAT (3X,I6,16X,F16.4,I16)                    DPRI 580
106 FORMAT (3X,I6,16X,F16.4)                        DPRI 590
107 FORMAT (3X,I3, 3X,F16.4)                        DPRI 600
C=====DPRI 610
END                                                  DPRI 620

```

```

SUBROUTINE INITEM                                INIT 10
INCLUDE COMDIM                                  INIT 20
C=====INIT 30
CB ASSIGN THE INITIAL TEMPERATURE DISTRIBUTION  INIT 40
C=====INIT 50
C INITEM1 THE TEMPERATURE BLOCK'S LOWER RADIAL-X BOUNDARY DOES INIT 60
C NOT COINCIDE WITH ANY OF THE RADIAL-X GRID BOUNDARIES INIT 70
C INITEM2 THE TEMPERATURE BLOCK'S UPPER RADIAL-X BOUNDARY DOES INIT 80
C NOT COINCIDE WITH ANY OF THE RADIAL-X GRID BOUNDARIES INIT 90
C INITEM3 THE TEMPERATURE BLOCK'S LOWER AXIAL-Z BOUNDARY DOES INIT 100
C NOT COINCIDE WITH ANY OF THE AXIAL-Z GRID BOUNDARIES INIT 110
C INITEM4 THE TEMPERATURE BLOCK'S UPPER AXIAL-Z BOUNDARY DOES INIT 120
C NOT COINCIDE WITH ANY OF THE AXIAL-Z GRID BOUNDARIES INIT 130
C INITEM5 THE TEMPERATURE BLOCK'S LOWER THETA-Y BOUNDARY DOES INIT 140
C NOT COINCIDE WITH ANY OF THE THETA-Y GRID BOUNDARIES INIT 150
C INITEM6 THE TEMPERATURE BLOCK'S UPPER THETA-Y BOUNDARY DOES INIT 160
C NOT COINCIDE WITH ANY OF THE THETA-Y GRID BOUNDARIES INIT 170
C INITEM7 AN INITIAL TEMPERATURE HAS NOT BEEN ASSIGNED TO SOME INIT 180
C INTERNAL POINT. INIT 190
C=====INIT 200
C=====INIT 210
C SET ALL INTERNAL TEMPERATURES TO ZERO ABSOLUTE INIT 220
DO 10 I=2,IM INIT 230
DO 10 J=2,JM INIT 240
DO 10 K=2,KM INIT 250
TT(I,J,K)=0.0 INIT 260
10 CONTINUE INIT 270
C=====INIT 280
C SET ALL EXTERNAL RADIAL-X COOLANT TEMPERATURES TO ZERO FAHRENHEIT INIT 290
DO 20 J=1,JMAX INIT 300
DO 20 K=1,KMAX INIT 310
TT(1,J,K)=460.0 INIT 320
TT(IMAX,J,K)=460.0 INIT 330
20 CONTINUE INIT 340
C=====INIT 350
C SET ALL EXTERNAL AXIAL-Z COOLANT TEMPERATURES TO ZERO FAHRENHEIT INIT 360
DO 30 I=1,IMAX INIT 370
DO 30 K=1,KMAX INIT 380
TT(I,1,K)=460.0 INIT 390
TT(I,JMAX,K)=460.0 INIT 400
30 CONTINUE INIT 410
C=====INIT 420
C SET ALL EXTERNAL THETA-Y COOLANT TEMPERATURES TO ZERO FAHRENHEIT INIT 430
DO 40 I=1,IMAX INIT 440
DO 40 J=1,JMAX INIT 450

```

```

      TT(I,J,1)=460.0
      TT(1,J,KMAX)=460.0
40 CONTINUE
C=====INIT 460
C=====INIT 470
C=====INIT 480
C=====INIT 490
C=====INIT 500
C READ AND CONVERT THE GEOMETRIC DATA OF A TEMPERATURE BLOCK
45 READ(5,100)RMIN,RMAX,ZMIN,ZMAX,TMIN,TMAX,FLAG1,FLAG2
C=====INIT 510
C HAVE ALL INITIAL TEMPERATURES BEEN READ? -YES,NO-
IF (RMIN+RMAX+ZMIN+ZMAX+TMIN+TMAX.EQ.0.0) GO TO 1000
C=====INIT 520
CALL FCARD (6.0,RMIN,RMAX,ZMIN,ZMAX,TMIN,TMAX,FLAG1,FLAG2)
C=====INIT 530
C=====INIT 540
C=====INIT 550
C=====INIT 560
C CONVERT THE TEMPERATURE BLOCK DATA TO THE CORRECT UNITS
IF (RMIN .LE. 0.0) RMIN = 1.0E-5
RMIN=RMIN/12.0
IF (RMAX .LE. 0.0) RMAX = 1.0E-5
RMAX=RMAX/12.0
ZMIN=ZMIN/12.0
ZMAX=ZMAX/12.0
TMIN=TMIN/SCALE
TMAX=TMAX/SCALE
C=====INIT 570
C=====INIT 580
C=====INIT 590
C=====INIT 600
C=====INIT 610
C=====INIT 620
C=====INIT 630
C=====INIT 640
C=====INIT 650
C=====INIT 660
CB ASSIGN THE UPPER AND LOWER RADIAL-X LIMITS OF THE TEMPERATURE BLOCK
C/ FIND THE INDEX OF THE LOW RADIAL-X TEMPERATURE BLOCK BOUNDARY
DO 50 I=1,IM
IF(RMIN.LE.1.00001*RL(I).AND.RMIN.GE.0.99999*RL(I))GO TO 60
50 CONTINUE
CE *****INIT 670
C THE LOW RADIAL-X BOUNDARY LIES OUTSIDE THE SYSTEM
CALL ERROR(7HINITEM1)
C=====INIT 680
C=====INIT 690
C=====INIT 700
C=====INIT 710
C=====INIT 720
C=====INIT 730
C=====INIT 740
C=====INIT 750
C ASSIGN THE INDEX OF THE LOW RADIAL-X TEMPERATURE BLOCK BOUNDARY
DO IMI=I+1
I1=I
C/ FIND THE INDEX OF THE HIGH RADIAL-X TEMPERATURE BLOCK BOUNDARY
DO 70 I=I1,IM
IF(RMAX.LE.1.00001*RL(I).AND.RMAX.GE.0.99999*RL(I))GO TO 80
70 CONTINUE
CE *****INIT 760
C THE HIGH RADIAL-X BOUNDARY LIES OUTSIDE THE SYSTEM
CALL ERROR(7HINITEM2)
C=====INIT 770
C=====INIT 780
C=====INIT 790
C=====INIT 800
C=====INIT 810
C=====INIT 820
C=====INIT 830
C=====INIT 840
C=====INIT 850
C=====INIT 860
C ASSIGN THE INDEX OF THE HIGH RADIAL-X TEMPERATURE BLOCK BOUNDARY
DO IMA=I
C ADJUST THE APPROPRIATE INDEX IF THE BLOCK DESCRIBES AN EXTERNAL
C .COOLANT
IF (IMA.EQ.1) IMI=1
C=====INIT 870
C=====INIT 880
C=====INIT 890
C=====INIT 900
C=====INIT 910

```

```

IF (IMI, EQ, IMAX) IMA=IMAX INIT 920
CE *****INIT 930
CB ASSIGN THE UPPER AND LOWER AXIAL-Z LIMITS OF THE TEMPERATURE BLOCK INIT 940
C/ FIND THE INDEX OF THE LOW AXIAL-Z TEMPERATURE BLOCK BOUNDARY INIT 950
DO 110 J=1, JM INIT 960
IF (ZMIN, LE, 1.00001*ZL(J), AND, ZMIN, GE, 0.99999*ZL(J)) GO TO 120 INIT 970
110 CONTINUE INIT 980
CE *****INIT 990
C THE LOW AXIAL-Z BOUNDARY LIES OUTSIDE THE SYSTEM INIT 1000
CALL ERROR(7HINITEM3) INIT 1010
=====INIT 1020
C ASSIGN THE INDEX OF THE LOW AXIAL-Z TEMPERATURE BLOCK BOUNDARY INIT 1030
120 JMI=J+1 INIT 1040
J1=J INIT 1050
C/ FIND THE INDEX OF THE HIGH AXIAL-Z TEMPERATURE BLOCK BOUNDARY INIT 1060
DO 130 J=J1, JM INIT 1070
IF (ZMAX, LE, 1.00001*ZL(J), AND, ZMAX, GE, 0.99999*ZL(J)) GO TO 140 INIT 1080
130 CONTINUE INIT 1090
CE *****INIT 1100
C THE HIGH AXIAL-Z BOUNDARY LIES OUTSIDE THE SYSTEM INIT 1110
CALL ERROR(7HINITEM4) INIT 1120
C ASSIGN THE INDEX OF THE HIGH AXIAL-Z TEMPERATURE BLOCK BOUNDARY INIT 1130
140 JMA=J INIT 1140
C ADJUST THE APPROPRIATE INDEX IF THE BLOCK DESCRIBES AN EXTERNAL INIT 1150
C COOLANT INIT 1160
IF (JMA, EQ, 1) JMI=1 INIT 1170
IF (JMI, EQ, JMAX) JMA=JMAX INIT 1180
CE *****INIT 1190
CB ASSIGN THE UPPER AND LOWER THETA-Y LIMITS OF THE TEMPERATURE BLOCK INIT 1200
C/ FIND THE INDEX OF THE LOW THETA-Y TEMPERATURE BLOCK BOUNDARY INIT 1210
DO 150 K=1, KM INIT 1220
IF (TMIN, LE, 1.00001*TL(K), AND, TMIN, GE, 0.99999*TL(K)) GO TO 160 INIT 1230
150 CONTINUE INIT 1240
CE *****INIT 1250
C THE LOW THETA-Y BOUNDARY LIES OUTSIDE THE SYSTEM INIT 1260
CALL ERROR(7HINITEM5) INIT 1270
=====INIT 1280
C ASSIGN THE INDEX OF THE LOW THETA-Y TEMPERATURE BLOCK BOUNDARY INIT 1290
160 KMI=K+1 INIT 1300
K1=K INIT 1310
C/ FIND THE INDEX OF THE HIGH THETA-Y TEMPERATURE BLOCK BOUNDARY INIT 1320
DO 170 K=K1, KM INIT 1330
IF (TMAX, LE, 1.00001*TL(K), AND, TMAX, GE, 0.99999*TL(K)) GO TO 180 INIT 1340
170 CONTINUE INIT 1350
CE *****INIT 1360
C THE HIGH THETA-Y BOUNDARY LIES OUTSIDE THE SYSTEM INIT 1370

```

```

      CALL ERROR(7HINITEM6)
C      ASSIGN THE INDEX OF THE HIGH THETA-Y TEMPERATURE BLOCK BOUNDARY
180 KMA=K
C      ADJUST THE APPROPRIATE INDEX IF THE BLOCK DESCRIBES AN EXTERNAL
C      COOLANT
      IF (KMA.EQ.1) KMI=1
      IF (KMI.EQ.KMAX) KMA=KMAX
CE *****
C      READ AND CONVERT THE TEMPERATURE DATA OF A TEMPERATURE BLOCK
      READ(5,101)TEM,FLAG1,FLAG2
      CALL FCARD (1.0,TEM,FLAG1,FLAG2,DUM,DUM,DUM,DUM,DUM)
      TEM=TEM+400.0
C=====
C      ASSIGN THE TEMPERATURES TO ALL POINTS IN THE TEMPERATURE BLOCK
      DO 200 I=MI,IMA
      DO 200 J=MI,JMA
      DO 200 K=MI,KMA
      TT(I,J,K)=TEM
      200 CONTINUE
      GO TO 45
C=====
C/ HAS A TEMPERATURE BEEN ASSIGNED TO EACH INTERNAL POINT? -NO,YES,-
1000 NC=NC+1
      CARL(NC)=0.0
      DO 1010 I=2,IM
      DO 1010 J=2,JM
      DO 1010 K=2,KM
      IF(TT(I,J,K).EQ.0.0) GO TO 1100
      1010 CONTINUE
CE *****
C      PUT THE INITIAL TEMPERATURES IN BOTH TEMPERATURE ARRAYS
      DO 1020 I=1,IMAX
      DO 1020 J=1,JMAX
      DO 1020 K=1,KMAX
      T(I,J,K)=TT(I,J,K)
      1020 CONTINUE
      RETURN
C=====
C      SOME INTERNAL POINT WAS NOT ASSIGNED AN INITIAL TEMPERATURE
1100 CALL ERROR(7HINITEM7)
C=====
      100 FORMAT(6E12.4,A6,A2)
      101 FORMAT(L12.4,60X,A6,A2)
C=====
      END

```

```

SUBROUTINE FLOWCA
INCLUDE COMDIM
=====FLOW 10
C=====FLOW 20
C READ AND PRINT THE COOLANT DATA CARDS FLOW 30
C=====FLOW 40
C ERROR STOPS= FLOW 50
C FLOWCA1 THE REYNOLDS NUMBER LIMITS ARE NOT IN SEQUENCE FLOW 60
C FLOWCA2 THE FLOWRATE LIMITS ARE NOT IN SEQUENCE FLOW 70
C FLOWCA3 THE INLET TEMPERATURE LIMITS ARE NOT IN SEQUENCE FLOW 80
C=====FLOW 90
C=====FLOW 100
C=====FLOW 110
DIMENSION CDIRCT( 2) FLOW 120
DATA CDIRCT /DHPOSITI, 6HNEGATI/ FLOW 130
C=====FLOW 140
WRITE (6,120)(ZA(I), I = 1,12) FLOW 150
120 FORMAT (1H1,30X,12A6,///,50X,23HCOOLANT SPECIFICATIONS) FLOW 160
C MAKE ALL COOLANTS NON-EXISTING FLOW 170
DO 5 I=1,MAXFLO FLOW 180
FLOW(I)= 0.0 FLOW 190
5 CONTINUE FLOW 200
C READ THE COOLANT INFORMATION FLOW 210
10 READ (5,100) DA,D1,D2,D3,D4,FLAG1,FLAG2 FLOW 220
NEUM+0.1 FLOW 230
C HAVE ALL COOLANT DATA CARDS BEEN READ? =,YES,NO=- FLOW 240
IF (N.LE.0) GO TO 291 FLOW 250
CALL FCARD (5.0,DA,D1,D2,D3,D4,FLAG1,FLAG2,DUM) FLOW 260
C RECORD THE EXISTENCE OF THE CURRENT COOLANT FLOW 270
FLOW(N)= 1.0 FLOW 280
WRITE (6,101) N FLOW 290
101 FORMAT (///,27H SPECIFICATIONS FOR COOLANT,13,/) FLOW 300
IP = IABS(IPATH(N)) FLOW 310
DIRECT = CDIRCT(1) FLOW 320
IF (IPATH(N) .GT. 0) GO TO 300 FLOW 330
DIRECT = CDIRCT(2) FLOW 340
300 WRITE (6,302) DIRECT, PNAME(IP) FLOW 350
302 FORMAT(31H THE COOLANT IS FLOWING IN THE ,A6,3HVE ,A6,10H DIRECTIOFLOW 360
-N) FLOW 370
C *****FLOW 380
C READ THE REMAINING SPECIFICATIONS OF THIS COOLANT FLOW 390
RLIM1(N)=D1 FLOW 400
RLIM2(N)=D2 FLOW 410
RLIM3(N)=D3 FLOW 420
RLIM4(N)=D4 FLOW 430
C/ CHECK TO SEE THAT THE REYNOLDS NUMBER LIMITS ARE IN SEQUENCE FLOW 440
IF (RLIM2(N).LT.RLIM1(N)) CALL ERROR (7HFLOWCA1) FLOW 450

```



```

IF(RLIM3(N).LT.RLIM2(N)) CALL ERROR (7HFLOWCA1)          FLOW 460
IF(RLIM4(N).LT.RLIM3(N)) CALL ERROR (7HFLOWCA1)          FLOW 470
CE *****FLOW 480
  WHILE(6,105)D1,D2,D3,D4          FLOW 490
  READ (5,100)FLODEP(N),FLIM1(N),FLIM2(N),FLIM3(N),      FLIM4(N) FLOW 500
  CALL FCARD (5.0,FLODEP(N),FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N), FLOW 510
  XFLAG1,FLAG2,DUM)          FLOW 520
C/ CHECK TO SEE THAT THE FLOWRATE LIMITS ARE IN SEQUENCE FLOW 530
  IF(FLIM2(N).LT.FLIM1(N))CALL ERROR (7HFLOWCA2)          FLOW 540
  IF(FLIM3(N).LT.FLIM2(N))CALL ERROR (7HFLOWCA2)          FLOW 550
  IF(FLIM4(N).LT.FLIM3(N))CALL ERROR (7HFLOWCA2)          FLOW 560
CE *****FLOW 570
  READ (5,100)TINDEP(N),TLIM1(N),TLIM2(N),TLIM3(N),      TLIM4(N) FLOW 580
  CALL FCARD (5.0,TINDEP(N),TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N), FLOW 590
  XFLAG1,FLAG2,DUM)          FLOW 600
C/ CHECK TO SEE THAT THE INLET TEMPERATURE LIMITS ARE IN SEQUENCE FLOW 610
  IF(TLIM2(N).LT.TLIM1(N))CALL ERROR(7HFLOWCA3)          FLOW 620
  IF(TLIM3(N).LT.TLIM2(N))CALL ERROR(7HFLOWCA3)          FLOW 630
  IF(TLIM4(N).LT.TLIM3(N))CALL ERROR(7HFLOWCA3)          FLOW 640
CE *****FLOW 650
  IFLO(N)=FLODEP(N)+1.01          FLOW 660
  IIN(N)=TINDEP(N)+1.01          FLOW 670
  I=IFLO(N)          FLOW 680
CB PRINT THE FLOWRATE AND INLET TEMPERATURE DEPENDENCES FLOW 690
C DETERMINE THE DEPENDENCE OF THE FLOWRATE -'NONE,TIME,OUTLET,INLET' FLOW 700
  GO TO (20,21,22,23),I          FLOW 710
C PRINT THAT THERE ARE NO STEPCHANGES IN FLOWRATE          FLOW 720
20 WHILE(6,106)          FLOW 730
  GO TO 30          FLOW 740
C PRINT THAT TIME DEFINES THE STEPCHANGES IN FLOWRATE          FLOW 750
21 WRITE(6,107)FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N)          FLOW 760
  GO TO 30          FLOW 770
C PRINT THAT THE OUTLET TEMPERATURE DEFINES THE STEPCHANGES IN FLOW 780
C ,FLOWRATE          FLOW 790
22 WHILE(6,108)FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N)          FLOW 800
  GO TO 30          FLOW 810
C PRINT THAT THE INLET TEMPERATURE DEFINES THE STEPCHANGES IN FLOW 820
C ,FLOWRATE          FLOW 830
23 WHILE(6,109)FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N)          FLOW 840
C/ DETERMINE THE DEPENDENCE OF THE INLET TEMPERATURE          FLOW 850
C -'NONE,TIME,FLOW,OUTLET'-          FLOW 860
30 I=IIN(N)          FLOW 870
  GO TO (31,32,33,34),I          FLOW 880
CE *****FLOW 890
C PRINT THAT THERE ARE NO STEPCHANGES IN THE INLET TEMPERATURE FLOW 900
31 WHILE(6,110)          FLOW 910

```

```

GO TO 50 FLOW 920
C PRINT THAT TIME DEFINES THE STEPCHANGES IN THE INLET TEMPERATURE FLOW 930
32 WRITE(6,111)TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N) FLOW 940
GO TO 50 FLOW 950
C PRINT THAT THE FLOWRATE DEFINES THE STEPCHANGES IN THE INLET FLOW 960
C .TEMPERATURE FLOW 970
33 WRITE(6,112)TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N) FLOW 980
GO TO 50 FLOW 990
C PRINT THAT THE OUTLET TEMPERATURE DEFINES THE STEPCHANGES IN FLOW,000
C .THE INLET TEMPERATURE FLOW1010
34 WRITE(6,113)TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N) FLOW1020
CE ***** FLOW1030
C DOES THE INLET OR OUTLET TEMPERATURE DEFINE THE STEPCHANGES FLOW1040
C .IN FLOWRATE? -'NO,YES'- FLOW1050
C/ DETERMINE THE DEPENDENCE OF THE INLET TEMPERATURE FLOW1060
C .-'NONE,TIME,FLOW,OUTLET'- FLOW1070
CE ***** FLOW1080
50 IF (FLODEP(N)-2.0)52,51,51 FLOW1090
C CONVERT THE STEPCHANGE LIMITS TO DEGREES RANKINE FLOW1100
51 FLIM1(N)=FLIM1(N)+460.0 FLOW1110
FLIM2(N)=FLIM2(N)+460.0 FLOW1120
FLIM3(N)=FLIM3(N)+460.0 FLOW1130
C DOES THE OUTLET TEMPERATURE DEFINE THE STEPCHANGES IN INLET FLOW1140
C .TEMPERATURE? -'NO,YES'- FLOW1150
52 IF (TINDEP(N).NE.3.0) GO TO 10 FLOW1160
C CONVERT THE STEPCHANGE LIMITS TO DEGREES RANKINE FLOW1170
53 TLIM1(N)=TLIM1(N)+460.0 FLOW1180
TLIM2(N)=TLIM2(N)+460.0 FLOW1190
TLIM3(N)=TLIM3(N)+460.0 FLOW1200
TLIM4(N)=TLIM4(N)+460.0 FLOW1210
GO TO 10 FLOW1220
291 NC=NC+1 FLOW1230
CARC(NC)=0.0 FLOW1240
DO 292 I=1,MAXFLO FLOW1250
292 NFLO(I)=0 FLOW1260
N=0 FLOW1270
DO 293 I=1,MAXFLO FLOW1280
IF (FLOW(I) .EQ. 0.0) GO TO 293 FLOW1290
N=N+1 FLOW1300
NFLO(N)=I FLOW1310
293 CONTINUE FLOW1320
RETURN FLOW1330
C=====FLOW1340
100 FORMAT (5E12.4,12X,A6,A2) FLOW1350
105 FORMAT (31H THE REYNOLDS NUMBER LIMITS ARE,1P4E14.4) FLOW1360
106 FORMAT (24H NO STEP CHANGES IN FLOW) FLOW1370

```

```

107 FORMAT (42H TIME(HR) DEFINES THE STEP CHANGES IN FLOW ,1P4E14.4) FLOW1380
108 FORMAT (55H OUTLET TEMPERATURE(F) DEFINES THE STEP CHANGES IN FLOWFLOW1390
X,1P4E14.4) FLOW1400
109 FORMAT(55H INLET TEMPERATURE(F) DEFINES THE STEP CHANGES IN FLOW, FLOW1410
X1P4E14.4) FLOW1420
110 FORMAT (37H NO STEP CHANGES IN INLET TEMPERATURE) FLOW1430
111 FORMAT (55H TIME(HR) DEFINES THE STEP CHANGES IN INLET TEMPERATUREFLOW1440
X,1P4E14.4) FLOW1450
112 FORMAT (62H FLOWRATE(LB/HR) DEFINES THE STEP CHANGES IN INLET TEMPFLOW1460
XERATURE,1P4E14.4) FLOW1470
113 FORMAT (66H OUTLET TEMPERATURE(F) DEFINES THE STEP CHANGES IN INLEFLOW1480
XT TEMPERATURE,1P4E14.4) FLOW1490
C=====FLOW1500
END FLOW1510

```

```

SUBROUTINE TIME                                     TIME 10
INCLUDE      COMDIM                                 TIME 20
C=====
C READ AND PRINT THE TIME STEPS                     TIME 30
C=====
C ERROR STOPS=                                       TIME 60
C   TIME1      THE FIRST TIME CARD DOES NOT SPECIFY TIME UNITS.   TIME 70
C               THE PROBLEM IS NOT AFFECTED IF THE STEADY STATE   TIME 80
C               OPTION IS BEING USED.                             TIME 90
C   TIME2      TOO MANY TIME HISTORY CARDS HAVE BEEN READ.       TIME 100
C   TIME3      THE INPUT CONTAINS NO TIME HISTORY CARDS AND THE   TIME 110
C               STEADY STATE OPTION IS NOT BEING USED.           TIME 120
C=====
C DIMENSION      PTIME ( 20,4)                          TIME 150
C EQUIVALLNCE    (RCP ,PTIME )                          TIME 160
C=====
C M=0                                                  TIME 190
C READ THE NEXT TIMESTEP                               TIME 200
  2 N=N+1                                             TIME 210
  READ(5,100)FTIME(M),DTIME(M),ST,FT,FLAG1,FLAG2    TIME 220
100 FORMAT (4E12.4,24X,A6,A2)                       TIME 230
  IF (DTIME(M) .LE. 0.0) GO TO 185                   TIME 240
  CALL FCARD (4.0,FTIME(M),DTIME(M),ST,FT,FLAG1,FLAG2,DUM,DUM) TIME 250
  ITAPE(M)=ST                                        TIME 260
  IUNIT=FT                                          TIME 270
C/ ARE THE UNITS THE SAME AS ON THE LAST CARD? -'NO,YES,WRONG'- TIME 280
  4 IF(IUNIT)5,5,7                                  TIME 290
  5 IF(N-1)6,6,61                                    TIME 300
CE *****
C THE FIRST TIMECARD HAS NO UNITS                    TIME 320
  6 CALL ERROR (SHTIME1)                             TIME 330
C/ CONVERT THE TIMES TO THE PROPER UNITS              TIME 340
  7 IF(IUNIT=2)8,9,60                                TIME 350
  8 D=3600.0                                         TIME 360
  GO TO 61                                           TIME 370
  9 D=60.0                                           TIME 380
CE *****
C CONVERT FTIME AND DTIME TO HOURS.                  TIME 400
  61 DTIME(M)=DTIME(M)/D                             TIME 410
  FTIME(M)=FTIME(M)/D                               TIME 420
C COMPENSATE FOR THE ROUND OFF ERROR IN DTIME.       TIME 430
  60 DTIME(M)=DTIME(M)+TIMERR(DTIME(M))             TIME 440
C HAVE ALL TIMECARDS BEEN READ? -'NO,YES'-          TIME 450

```

```

GO TO 2
185 IF(.NOT.SW(10).AND.M.EQ.1) CALL ERROR(5HTIME3)
NC=NC+1
CARD(NC)=0.0
MMAA=M-1
C ARE THERE TOO MANY TIMECARDS -,NO,YES'-
IF(MMAX-20)64,64,63
C TOO MANY DIFFERENT TIMESTEPS ARE USED
63 CALL ERROR(5HTIME2)
64 IF(SW(10).AND.MMAX.EQ.0) GO TO 250
C=====
C PRINT THE TIME STEPS AND ITERATION PARAMETERS
DO 65 M=1,MMAX
PTIME(M,1) = FTIME(M) * 60.0
PTIME(M,2) = FTIME(M) * 3600.0
PTIME(M,3) = DTIME(M) * 60.0
65 PTIME(M,4) = DTIME(M) * 3600.0
WRITE(6,102) (ZA(I), I = 1,12)
102 FORMAT (1H1,30X,12A6,///,50X,13HTIME HISTORY,/,10X,1AHEND OF TIME
- PERIOD, 33X,9HTIME STEP,23X,15HPRINT FREQUENCY/7X,5HHOURS,6X,
-7HMINUTES,6X,7HSECONDS,15X,5HHOURS,6X,7HMINUTES,6X,7HSECONDS//)
WRITE(6,101) (FTIME(M),PTIME(M,1),PTIME(M,2),DTIME(M),PTIME(M,3),
-PTIME(M,4),ITAPE(M), M = 1,MMAX)
101 FORMAT (1H ,1P3E13.6,7X,3E13.6,14)
IF(.NOT.SW(10)) RETURN
WRITE(6,240)
240 FORMAT(1HU, 2BH THE ABOVE TIME HISTORY WILL BE IGNORED BECAUSE
THE STEADY STATE OPTION IS BEING USED.)
250 DO 260 M=1,21
DTIME(M)=0.0
FTIME(M)=0.0
260 ITAPE(M)=0
MMAA=1
RETURN
C *****
C *****
C *****
FUNCTION TIMERR(TIM)
C =====
C DETERMINE MAXIMUM ROUND OFF ERROR INCURRED IN CONVERTING TIME SCALE
C DATA TO HOURS.
C =====
C REAL MAG
C =====
C DIMENSION MAG(31)
DATA(MAG(I),I=1,31) / 1.0E-15,1.0E-14,1.0E-13,1.0E-12,1.0E-11,

```

```

.          1.0E-10,1.0E-9,1.0E-8,1.0E-7,1.0E-6,1.0E-5, TIME 920
.          1.0E-4,1.0E-3,1.0E-2,1.0E-1,1.0,1.0E1,1.0E2, TIME 930
.          1.0E3,1.0E4,1.0E5,1.0E6,1.0E7,1.0E8,1.0E9, TIME 940
.          1.0E10,1.0E11,1.0E12,1.0E13,1.0E14,1.0E15 / TIME 950
C =====TIME 960
  TIMLRR=0. TIME 970
  IF(TIM.LT.MAG(1).OR.TIM.LT.0.) RETURN TIME 980
  ISUM=-16 TIME 990
  DO 100 I=1,30 TIME1000
  ISUM=ISUM+1 TIME1010
  IF(TIM.GT.MAG(I).AND.TIM.LT.MAG(I+1)) GO TO 110 TIME1020
100 CONTINUE TIME1030
  RETURN TIME1040
110 KX=ISUM-7 TIME1050
  TIMLRR= MAG(16+KX) TIME1060
C =====TIME1070
  RETURN TIME1080
  END TIME1090

```

```

SUBROUTINE EXTRA                                EXTR 10
  INCLUDE COMDIM                                EXTR 20
C=====EXTR 30
C  READ THE EXTRA FUNCTION COEFFICIENTS AND PRINT THEM EXTR 40
C=====EXTR 50
C=====EXTR 60
  DIMENSION A (18)                              EXTR 70
  EQUIVALENCE (A(1),A1)                          EXTR 80
C=====EXTR 90
  READ (5,100) A1,A2,A3,A4,A5,A6,FLAG1,FLAG2     EXTR 100
  CALL FCARD (6.0,A1,A2,A3,A4,A5,A6,FLAG1,FLAG2) EXTR 110
  READ (5,100) A7,A8,A9,A10,A11,A12,FLAG1,FLAG2  EXTR 120
  CALL FCARD (6.0,A7,A8,A9,A10,A11,A12,FLAG1,FLAG2) EXTR 130
  READ (5,100) A13,A14,A15,A16,A17,A18,FLAG1,FLAG2 EXTR 140
  CALL FCARD (6.0,A13,A14,A15,A16,A17,A18,FLAG1,FLAG2) EXTR 150
  DO 110 I=1,18                                  EXTR 160
  IF (A(I) .EQ. 0.0) A(I)=0.0                    EXTR 170
110 CONTINUE                                     EXTR 180
  WRITE(6,101) (ZA(I), I = 1,12)                 EXTR 190
  WRITE (6,102) (I,A(I), I=1,18)                 EXTR 200
C  SPACE OVER BLANK CARD                          EXTR 210
  READ (5,100) X                                  EXTR 220
  NC=NC+1                                         EXTR 230
  CARD(NC)=0.0                                    EXTR 240
C=====EXTR 250
  100 FORMAT (6E12.4,A6,A2)                       EXTR 260
  101 FORMAT(1H1,30X,12A6,///,50X,28HFUNCTION CONTROL CONSTANTS,/,6X, EXTR 270
  -15HCONSTANT NUMBER,12X,5HVALUE/)              EXTR 280
  102 FORMAT (1H0,114,12X,1PE14.6)               EXTR 290
C=====EXTR 300
  RETURN                                          EXTR 310
  END                                             EXTR 320

```

```

SUBROUTINE PRETEM                                PRET 10
INCLUDE COMDIM                                  PRET 20
C=====PRET 30
C=====PRET 40
C* ERROR STOP:                                PRET 50
C* PRETEM1 THE PROBLEM SIZE DOES NOT MATCH THE INITIAL PRET 60
C* TEMPERATURE DISTRIBUTION DATA              PRET 70
C=====PRET 80
EQUIVALENCE (TEMP ,W )                        PRET 90
INTEGER TT                                     PRET 100
DIMENSION NUMBER( 50),TEMP ( 1)              PRET 110
DATA NUMBER /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19, PRET 120
- /20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35, PRET 130
- /36,37,38,39,40,41,42,43,44,45,46,47,48,49,50/ PRET 140
C=====PRET 150
WRITE (6,1105) (ZA(I), I = 1,12)              PRET 160
1105 FORMAT (1H1,30X,12A6,///,46X,32HINPUT TEMPERATURE DISTRIBUTION) PRET 170
C=====PRET 180
CURTI=0.0                                       PRET 190
NITER=0                                         PRET 200
C=====PRET 210
C READ THE FIRST CARD WITH THE GRID SIZE       PRET 220
READ (5,260) CURTI, BIMAX,BJMAX,BKMAX,BNITER,FLAG1,FLAG2 PRET 230
IAMAX=BIMAX +0.1                               PRET 240
JAMAX=BJMAX +0.1                               PRET 250
KAMAX=BKMAX +0.1                               PRET 260
NITER = BNITER +0.1                             PRET 270
C IS A PREVIOUSLY CALCULATED TEMPERATURE DISTRIBUTION PROVIDED? PRET 280
C .-'NO,YES='-                                  PRET 290
IF (IAMAX .NE. 0) GO TO 105                      PRET 300
NC=NC+1                                         PRET 310
CARD(NC)=0.0                                    PRET 320
GO TO 30                                         PRET 330
C=====PRET 340
C/ IS THE GRID SIZE CORRECT? .-'NO,YES='-      PRET 350
105 IF (IAMAX.NE.IMAX) GO TO 11                  PRET 360
IF (JAMAX.NE.JMAX) GO TO 11                    PRET 370
IF (KAMAX.EQ.KMAX) GO TO 21                    PRET 380
CE *****PRET 390
C THE TEMPERATURE INPUT DOES NOT MATCH THE PROBLEM PRET 400
11 CALL ERROR (7HPRETEM1)                       PRET 410
C=====PRET 420
C READ THE INITIAL TEMPERATURE DISTRIBUTION     PRET 430
21 CALL FCARD (5.0,CURTI,BIMAX,BJMAX,BKMAX,BNITER,FLAG1,FLAG2,DUM) PRET 440
NT=IMAX*JMAX*KMAX                               PRET 450

```



```

CALL RCARD (TEMP,NT)
NT=0
DO 125 K=1,KMAX
DO 125 J=1,JMAX
DO 125 I=1,IMAX
NT=NT+1
T(I,J,K)=TEMP(NT)
125 CONTINUE
CALL RCARD(TI,MAXFLO)
CALL RCARD(TO,MAXFLO)
C READ PAST BLANK CARD
READ (5,260) X
NC=NC+1
CARD(NC)=0.0
C=
=====
C PREPARE ALL COOLANT DATA FOR PRINTING
WRITE (6,108)
108 FORMAT (1H0,49X,28HCOOLANT TEMPERATURES (F)/17X,14HCOOLANT NUMBER,4X,5HINLET,5X,6HOUTLET,10X,14HCOOLANT NUMBER,4X,5HINLET,5X,6HOUTLET)
DO 150 N=1,MAXFLO,2
IF (NFLO(N).EQ.0) GO TO 180
N1=NFLO(N)
N2=NFLO(N+1)
ITI(N1)=TI(N1)-459.5
ITO(N1)=TO(N1)-459.5
IF (N2.EQ.0 .OR. N.EQ.15) GO TO 140
ITI(N2)=TI(N2)-459.5
ITO(N2)=TO(N2)-459.5
WRITE (6,160) N1,ITI(N1),ITO(N1),N2,ITI(N2),ITO(N2)
160 FORMAT (22X,I2,I10,I11,15X,I2,I16,I11)
GO TO 150
140 WRITE (6,160) N1,ITI(N1),ITO(N1)
150 CONTINUE
180 IF(.NOT.SW(10)) GO TO 185
CURT1=0.0
NITER=0
185 X1=CURT1+0.0
X2=CURT1+3600.0
WRITE (6,250) CURT1,X1,X2,NITER
250 FORMAT (1H0,21H THE CURRENT TIME IS,F10.4,11H HOURS = ,F10.4,13H MINUTES = ,F13.5,8H SECONDS,4X,I4,31H ITERATIONS HAVE BEEN PERFORMED,/)
C=====
C CONVERT TEMPERATURES TO INTEGRAL OUTPUT UNITS
38 DO 15 I=1,IMAX

```

```

PRET 460
PRET 470
PRET 480
PRET 490
PRET 500
PRET 510
PRET 520
PRET 530
PRET 540
PRET 550
PRET 560
PRET 570
PRET 580
PRET 590
PRET 600
PRET 610
PRET 620
PRET 630
PRET 640
PRET 650
PRET 660
PRET 670
PRET 680
PRET 690
PRET 700
PRET 710
PRET 720
PRET 730
PRET 740
PRET 750
PRET 760
PRET 770
PRET 780
PRET 790
PRET 800
PRET 810
PRET 820
PRET 830
PRET 840
PRET 850
PRET 860
PRET 870
PRET 880
PRET 890
PRET 900
PRET 910

```

```

DO 15 K=1,KMAX
DO 15 J=1,JMAX
TT(I,J,K) = T(I,J,K) - 459.5
C 15 CONTINUE
SET MESH KIBS EQUAL TO ZERO
DO 500 L = 1,JMAX
TT(1,L,1) = 0
TT(IMAX,L,1) = 0
TT(1,L,KMAX) = 0
500 TT(IMAX,L,KMAX) = 0
DO 525 L = 1,IMAX
TT(L,1,1) = 0
TT(L,JMAX,1) = 0
TT(L,1,KMAX) = 0
525 TT(L,JMAX,KMAX) = 0
DO 550 L = 1,KMAX
TT(1,1,L) = 0
TT(IMAX,1,L) = 0
TT(1,JMAX,L) = 0
550 TT(IMAX,JMAX,L) = 0
C=====
WRITE (6,201)
201 FORMAT (1H0,47X,20HTEMPERATURES (F))
CALL ARRAY1 (TT,DUMM,1)
1000 RETURN
C=====
110 FORMAT (E10.0,4I10)
200 FORMAT (E12.6,4F12.1,12X,A6,A2)
122 FORMAT (////)
C* *****PRET1210
C* *****PRET1220
SUBROUTINE RCARD (TR,LENGTH)
C= =====PRET1240
DIMENSION TR ( 1),TC ( 6)
C= =====PRET1250
N=LENGTH/6
NLEFT=LENGTH-N*6
N2=0
PRET1290
DO 100 I=1,N
PRET1300
N1=N2+1
PRET1310
N2=N2+6
PRET1320
REAL (5,110) (TR(J),J=N1,N2),FLAG1,FLAG2
PRET1330
CALL FCARD (6.0,TR(N1),TR(N1+1),TR(N1+2),TR(N1+3),TR(N1+4),TR(N1+5)
PRET1340
1),FLAG1,FLAG2)
PRET1350
100 CONTINUE
PRET1360
IF (NLEFT.EQ.0) GO TO 130
PRET1370

```

```
READ (5,110) (TC(I),I=1,6),FLAG1,FLAG2          PRET13A0
CALL FCARD(6,0,TC(1),TC(2),TC(3),TC(4),TC(5),TC(6),FLAG1,FLAG2) PRET1390
DO 120 I=1,NLEFT                                  PRET1400
J=N2+1                                             PRET1410
TR(J)=TC(I)                                       PRET1420
120 CONTINUE                                       PRET1430
130 RETURN                                         PRET1440
110 FORMAT (6E12.4,A6,A2)                         PRET1450
END                                                PRET1460
```

```

SUBROUTINE MP2
INCLUDE COMDIM
C =====
CB CALCULATE THE TEMPERATURE DISTRIBUTION AS A FUNCTION OF TIME
C =====
C ERROR STOPS=
C MP2 1 THE CURRENT TIME OF THE PREVIOUSLY PUNCHED
C TEMPERATURE DISTRIBUTION IS GREATER THAN THE ENDING
C TIME OF ANY GIVEN TIMESTEP.
C =====
LOGICAL NOTEST
DIMENSION IMP2 ( 2)
DATA IMP2 /6IMP2 1, 6H /
C =====
CS1=0.0
CS2=0.0
DPE=.FALSE.
IWN=10
FIRST=.TRUE.
NSTORE=3
C= =====
C INITIALIZE THE CAPACITANCES.
DO 110 K=1,KMAX
DO 110 J=1,JMAX
DO 110 I=1,IMAX
RCP(I,J,K)=1.0E12
110 CONTINUE
C= =====
C THE CODE COMPLETES ONE ENTIRE SET OF CALCULATIONS WITH A TIME STEP
C OF 1.0E-10 SECONDS TO INITIALIZE THE PROBLEM. THE RESULTS OF THIS
C ITERATION ARE NOT PRINTED OR IS THE ITERATION COUNTED. THEN THE
C USER SPECIFIED ITERATIONS ARE PERFORMED.
C INSERT THE INITIALIZATION ITERATION IN THE PROPER PLACE IN THE
C USER SPECIFIED ITERATIONS
IF (CURTI.EQ.0.0) GO TO 130
DO 120 I=1,IMAX
IF (CURTI.LT.FTIME(I)) GO TO 140
120 CONTINUE
IERROR(1)=IMP2(1)
IERROR(2)=IMP2(2)
CALL ERROR2
130 I=1
140 NITER=NITER+1
M=I-1
K=IMAX

```

150	FTIME(K+1)=FTIME(K)	MP2	460
	DTIME(K+1)=DTIME(K)	MP2	470
	ITAPE(K+1)=ITAPE(K)	MP2	480
	K=K+1	MP2	490
	IF (K.GE.I) GO TO 150	MP2	500
	FTIME(I)=CURTI+1.0E-10/3600.0	MP2	510
	DTIME(I)=1.0E-10/3600.0	MP2	520
	ITAPE(I)=0	MP2	530
	MMAX=MMAX+1	MP2	540
	GO TO 165	MP2	550
C=	=====	MP2	560
160	FIRST=.FALSE.	MP2	570
	IF (.NOT.SW(10)) GO TO 165	MP2	580
	IF (CS1.LT.1.0) GO TO 162	MP2	590
	GO TO 165	MP2	600
162	CALL STEADY	MP2	610
	IWNS=NIITER+19	MP2	620
C	START THE NEXT TIME PERIOD	MP2	630
165	M=M+1	MP2	640
	CTI=CURTI	MP2	650
	IC=1	MP2	660
C	HAVE ALL THE TIME PERIODS BEEN FINISHED	MP2	670
	IF (M.GT.MMAX) GO TO 200	MP2	680
	ENDTI=FTIME(M)	MP2	690
	DTSIAN=DTIME(M)	MP2	700
	NTSIAN=ITAPE(M)	MP2	710
	DT=DTSIAN	MP2	720
	NTA=NTSIAN	MP2	730
	IF (NTSIAN.EQ.0) NTA=10000	MP2	740
C=	=====	MP2	750
CB	CALCULATE THE TEMPERATURES DURING ONE SERIES OF TIMESTEPS	MP2	760
C	.-'END,NEXT'.	MP2	770
C	HAS COMPUTER TIME ALMOST RUN OUT? -'NO,YES'.	MP2	780
170	IF (WARN(I)) 181, 200, 200	MP2	790
C	HAS NUMBER OF PRINTED PAGES SPECIFIED ALMOST RUN OUT? -'NO,YES'.	MP2	800
181	CALL PAGLIM(4201,5)	MP2	810
C	DEFINE THE TIME IN THE MIDDLE OF THE TIMESTEP	MP2	820
180	DATI=CURTI+DTSIAN/2.0	MP2	830
C	SAVE TEMPERATURE RESULTS OF NEXT TO LAST SMOOTHING ITERATION FOR	MP2	840
C	CALCULATING RESIDUALS(STEADY STATE OPTION ONLY).	MP2	850
	IF (.NOT.SW(10)) GO TO 182	MP2	860
	IF (NIITER.NI.IWNS) GO TO 182	MP2	870
	REWIND NSTORE	MP2	880
	WRITE (NSTORE)((T(I,J,K),I=1,IQ),J=1,JQ),K=1,KQ)	MP2	890
C	CALCULATE THE MATERIAL PROPERTIES	MP2	900
182	CALL BLOCK	MP2	910

```

C      CALCULATE THE AVERAGE CONDUCTIVITIES BETWEEN POINTS                MP2  920
      CALL CONDOC                                                            MP2  930
C      CALCULATE THE NEXT TEMPERATURE DISTRIBUTION                          MP2  940
      NOTEST=.TRUE.
      CALL STEP($183,NOTEST)                                                MP2  960
C      CALCULATE THE COOLANT TEMPERATURES                                  MP2  970
183    CALL COOL                                                            MP2  980
C      CALCULATE THE BOUNDARY TEMPERATURES.                                MP2  990
      CALL SURT                                                            MP2 1000
C=     =====MP2 1010
      NITER=NITER+1                                                         MP2 1020
      CURTI=C*TI+IC*DTSTAN                                                 MP2 1030
      NTA=NTA-1                                                             MP2 1040
C      SHOULD THE RESULTS OF THIS TIME STEP BE PRINTED ?                   MP2 1050
C      IF THIS IS THE LAST TIME STEP FOR THE CURRENT TIME PERIOD AND       MP2 1060
C      IF PRINTING HAS BEEN REQUESTED WITHIN ANY TIME PERIOD OF THE       MP2 1070
C      PROBLEM, THEN THE RESULTS OF THIS TIME PERIOD WILL BE PRINTED.     MP2 1080
      IF (CURTI.GE.ENDTI.AND).NTSTAN.NE.0) GO TO 185                       MP2 1090
      IF (NTA.GT.0) GO TO 190                                               MP2 1100
      NTA=NTSTAN                                                           MP2 1110
C      PRINT THE RESULTS OF THIS TIMESTEP                                   MP2 1120
185    CALL PRINT                                                            MP2 1130
C=     =====MP2 1140
C      HAS THIS SERIES OF TIMESTEPS BEEN COMPLETED? -'YES,NO'-           MP2 1150
190    IF (CURTI.GE.ENDTI) GO TO 160                                       MP2 1160
      IC = IC + 1                                                           MP2 1170
      GO TO 170                                                            MP2 1180
CE     *****MP2 1190
C      SET THE SWITCHES TO FORCE PRINT OF DEBUG INFORMATION AT THE         MP2 1200
C      END OF THE RUN ONLY                                                 MP2 1210
200    IF (SW(11)) SW(5)=.TRUE.                                             MP2 1220
      IF (SW(12)) SW(6)=.TRUE.                                             MP2 1230
      IF (SW(13)) SW(7)=.TRUE.                                             MP2 1240
      DP=.TRUE.                                                            MP2 1250
C      PRINT THE CURRENT TEMPERATURE DISTRIBUTION                           MP2 1260
      CALL PRINT                                                            MP2 1270
C=     =====MP2 1280
      IF (SW(3)) END FILE OUTTAP                                           MP2 1290
C=     =====MP2 1300
C      IS A PUNCHED TEMPERATURE DISTRIBUTION DESIRED? -'NO,YES'-         MP2 1310
      IF (.NOT.SW(2)) RETURN                                               MP2 1320
C      =====MP2 1330
C      PUNCH THE TEMPERATURE DISTRIBUTION                                  MP2 1340
C      =====MP2 1350
201    CALL PUN                                                            MP2 1360
      RETURN                                                                MP2 1370
      END                                                                    MP2 1380

```

```

SUBROUTINE ERROR2(IB)                                ERRO 10
  INCLUDE COMDIM                                    ERRO 20
C=====ERRO 30
C=====ERRO 40
C=====ERRO 50
C=====ERRO 60
  DIMENSION IB ( 2)                                ERRO 70
C=====ERRO 80
  IEKOR(1) = IB(1)                                  ERRO 90
  IEKOR(2) = IB(2)                                  ERRO 100
C  RECORD THE ERROR                                  ERRO 110
  WRITE(6,100) IEKOR                                ERRO 120
100 FORMAT(10H ERROR AT ,2A6)                       ERRO 130
C  IS THE DUMP SWITCH ON? -'YES,NO'                ERRO 140
C  GIVE A DUMP                                       ERRO 150
  IF (SW(1)) CALL DUMP                               ERRO 160
  STOP                                              ERRO 170
C=====ERRO 180
  END                                              ERRO 190

```

```

SUBROUTINE STEADY                                STEA 10
INCLUDE      COMDIM                              STEA 20
=====STEA 30
C CALCULATE THE STEADY STATE TEMPERATURES.      STEA 40
C =====STEA 50
LOGICAL      ENDSW, LREAD, WRRES, NOTEST, FIR2   STEA 60
DIMENSION    XT(IQ,JQ,KQ), STEADY(IQ,JQ,KQ)     STEA 70
EQUIVALENCE  (XT,TT), (RCP,STEADY)             STEA 80
=====STEA 90
C =====STEA 100
C SET CONSTANTS AND INDICATORS,                 STEA 110
C =====STEA 120
DATA = -1.0E+8                                  STEA 130
TOL=1.0E-4                                       STEA 140
RIFAC=1.50                                       STEA 150
DTMAX=3.0                                        STEA 160
ITMAX=1000                                       STEA 170
IPSL1=10                                         STEA 180
NSTORE=2                                         STEA 190
NSTORE=3                                         STEA 200
LSTORL=4                                         STEA 210
C SET AN INDICATOR TO SUPPRESS ANY FURTHER CALCULATION OF REAL
C CAPACITANCE TERMS.                           STEA 220
CS2=2.0                                          STEA 230
=====STEA 240
C =====STEA 250
C INITIALIZATIONS                              STEA 260
C =====STEA 270
CURT1=0.0                                        STEA 280
NITER=0                                          STEA 290
IDT=1                                           STEA 300
NSET=0                                          STEA 310
N=0                                              STEA 320
NT=0                                            STEA 330
DTFAC=1.0                                       STEA 340
FIRL=.FALSE.                                   STEA 350
NADV=1                                          STEA 360
LREAD=.TRUE.                                    STEA 370
ENDSW=.FALSE.                                  STEA 380
WRRES=.FALSE.                                  STEA 390
=====STEA 400
C PREPARE TO DO A SET OF ITERATIONS.           STEA 410
C =====STEA 420
C DETERMINE THE PARAMETERS WHICH CONTROL THE TIME STEP MAGNITUDE
C AND RATE OF INCREASE DURING THE CURRENT ITERATION SET. STEA 430
C =====STEA 440
230 IF(10(.EQ.2) GO TO 250                      STEA 450

```


	NADV=1	STE 460
	IF(FIR2) NADV=0	STE 470
	GO TO 270	STE 480
250	NADV=0	STE 490
	DTFAC=DTFAC/RIFAC	STE 500
	FIRL=.TRUE.	STE 510
C	IF INSTABILITIES OCCURRED DURING THE PRECEEDING ITERATION,AND IF	STE 520
C	THIS IS NOT THE FIRST ITERATION SET,RECOVER FROM DRUM THE	STE 530
C	TEMPERATURE RESULTS OF THE NEXT TO LAST SUCCESSFUL SET.	STE 540
	ISTORE=MSTORE	STE 550
	IF(LREAD) ISTORE=LSTORE	STE 560
	CALL STORE(ISTORE,1)	STE 570
	GO TO 270	STE 580
260	DTFAC=DTFAC/RIFAC	STE 590
	NADV=0	STE 600
C	IF INSTABILITY OCCURS DURING THE FIRST ITERATION SET,RECOVER THE	STE 610
C	INITIAL TEMPERATURES FROM DRUM.	STE 620
	CALL STORE(MSTORE,1)	STE 630
C	=====	STE 640
C	PERFORM A SET OF ITERATIONS.	STE 650
C	=====	STE 660
270	DO 1000 M=1,IPSET.	STE 670
C	DETERMINE THE TIME STEP FOR EACH ITERATION.DO NOT ALLOW THE TIME	STE 680
C	STEP TO EXCEED DTMAX HOURS.	STE 690
	XN=N	STE 700
	DTSTAN=DTFAC*(.10*XN+1.0)	STE 710
	IF(DTSTAN.GT.DTMAX) DTSTAN=DTMAX	STE 720
	DT=DTSTAN	STE 730
C	DETERMINE THE CONDUCTANCES AND PSEUDO CAPACITANCES FOR THE	STE 740
C	CURRENT ITERATION.	STE 750
	CALL BLOCK	STE 760
	CALL CONDOC	STE 770
	CALL STACAP	STE 780
C	IF THIS IS THE FIRST ITERATION OF A PROBLEM,SAVE THE INITIAL	STE 790
C	TEMPERATURES ON DRUM.	STE 800
	IF(NT.NE.0) GO TO 300	STE 810
	CALL STORE(MSTORE,2)	STE 820
C	ADVANCE THE ITERATION COUNTERS.	STE 830
300	N=N+NADV	STE 840
	NT=NT+1	STE 850
C	CALCULATE TEMPERATURES FOR THE CURRENT ITERATION.	STE 860
	IF(WARN(I).LT.0.0.AND.NT.LT.ITMAX) GO TO 400	STE 870
	ENDSW=.TRUE.	STE 880
	NOTEST=.TRUE.	STE 890
	CALL STEP(5410,NOTEST)	STE 900
	GO TO 410	STE 910

```

400 NOTLST=.FALSE. STEA 920
CALL STEP(320,NOTEST) STEA 930
410 CALL COOL STEA 940
CALL SUKT STEA 950
IF(LND5W) GO TO 420 STEA 960
C IF THIS IS THE NEXT TO LAST ITERATION OF THE SET,WRITE THE STEA 970
C TEMPERATURE RESULTS ON DRUM FOR USE IN CHECKING CONVERGENCE STEA 980
C AND STABILITY. STEA 990
IF (M.NE.IPSET-1) GO TO 1000 STEA,000
REWIND NSTORE STEA1010
WRITE(NSTORE) (((T(I,J,K),I=1,IQ),J=1,JQ),K=1,KQ) STEA1020
1000 CONTINUE STEA1030
C ADVANCE THE SET COUNTER. STEA1040
NSEIT=NSEIT+1 STEA1050
=====STEA1060
C WRITE TEMPERATURE RESULTS ON TWO DIFFERENT DRUM UNITS IN STEA1070
C ALTERNATING SEQUENCE AS THE SETS ARE COMPLETED,SET AN INDICATOR STEA1080
C TO SHOW WHICH DRUM UNIT CONTAINS THE OLDEST TEMPERATURES,DO NOT STEA1090
C WRITE ON DRUM UNITS IF THE TIME STEP HAS REACHED 1000 HOURS,THE STEA1100
C OLDEST OF THE TWO AVAILABLE SETS OF TEMPERATURES ARE USED TO STEA1110
C REPLACE ANY SET OF RESULTS WHICH CONTAINS EITHER NEGATIVE OR STEA1120
C EXTREMELY LARGE TEMPERATURES. STEA1130
=====STEA1140
IF(LREAD) GO TO 305 STEA1150
ISTORE=NSTORE STEA1160
LREAD=.TRUE. STEA1170
GO TO 310 STEA1180
305 ISTORE=LSTORE STEA1190
LREAD=.FALSE. STEA1200
310 CALL STORE(ISTORE,2) STEA1210
GO TO 320 STEA1220
=====STEA1230
C ENTER FROM SUBROUTINE PRINT TO EVALUATE RESIDUALS FOR THE FINAL STEA1240
C SMOOTHING ITERATION. STEA1250
C =====STEA1260
ENTRY RESID STEA1270
ARLS=.TRUE. STEA1280
C RECOVER THE TEMPERATURES WHICH WERE STORED ON DRUM FOR USE IN STEA1290
C CHECKING CONVERGENCE AND STABILITY,FOR THE ENTRY THROUGH RESID, STEA1300
C THESE TEMPERATURES WERE STORED ON DRUM IN SUBROUTINE MP2. STEA1310
320 REWIND NSTORE STEA1320
READ(NSTORE) (((XT(I,J,K),I=1,IQ),J=1,JQ),K=1,KQ) STEA1330
=====STEA1340
C EVALUATE THE PROXIMITY TO STEADY STATE AND DETERMINE THE CURRENT STEA1350
C VALUE OF THE STABILITY PARAMETER. STEA1360
C =====STEA1370

```

```

420 DO 1500 I=2,IM
DO 1500 J = 2,JM
DO 1500 K=2,KM
IF (RCP(I,J,K) .GT. 1.0E10) GO TO 1500
CAP = (RCP(I,J,K) * (T(I,J,K) - XT(I,J,K)))/DTSTAN
TCENT = ( T(I,J,K) + XT(I,J,K))/2.0
AA1=(TCENT - ((T(I+1,J,K) + XT(I+1,J,K))/2.0)
AA2=((T(I-1,J,K) + XT(I-1,J,K))/2.0) - TCENT
AA3=TCENT - ((T(I,J+1,K) + XT(I,J+1,K))/2.0)
AA4=((T(I,J-1,K) + XT(I,J-1,K))/2.0) - TCENT
AA5=TCENT - ((T(I,J,K+1) + XT(I,J,K+1))/2.0)
AA6=((T(I,J,K-1) + XT(I,J,K-1))/2.0) - TCENT
H1=-RR(I,J,K)*KR(I,J,K)*AA1
H2=NR(I-1,J,K)*KR(I-1,J,K)*AA2
H3=-KZ(I,J,K)*KZ(I,J,K)*AA3
H4=KZ(I,J-1,K)*KZ(I,J-1,K)*AA4
H5=-KT(I,J,K)*KT(I,J,K)*AA5
H6=KT(I,J,K-1)*KT(I,J,K-1)*AA6
VERG = H1 + H2 + H3 + H4 + H5 + H6 + W(I,J,K)
AVERG=(ABS(H1) + ABS(H2) + ABS(H3) + ABS(H4) + ABS(H5) + ABS(H6)
+ ABS(W(I,J,K)))
ACAP = ABS(CAP)
C CALCULATE THE CONVERGENCE PARAMETER AT EVERY MATERIAL POINT IN
C THE PROBLEM.
STEADY(I,J,K) = ACAP/AVERG
1500 CONTINUE
IF(LINDSW.OR.WRES) GO TO 360
C =====STEAI1650
C COMPARE THE CONVERGENCE PARAMETER AT EVERY MATERIAL POINT STEAI1660
C AGAINST A SPECIFIED MAXIMUM ALLOWABLE VALUE TO TEST WHETHER OR STEAI1670
C NOT THE STEADY STATE SOLUTION HAS BEEN ATTAINED. STEAI1680
C =====STEAI1690
330 DO 200 I=2,IM STEAI1700
DO 200 J=2,JM STEAI1710
DO 200 K=2,KM STEAI1720
IF(STEADY(I,J,K).GT.1.0E10) GO TO 200 STEAI1730
IF(STEADY(I,J,K).GE.TOL) GO TO 210 STEAI1740
200 CONTINUE STEAI1750
C =====STEAI1760
C PREPARE TO RETURN TO SUBROUTINE MP2 AND PERFORM THE SMOOTHING STEAI1770
C ITERATIONS. STEAI1780
C =====STEAI1790
205 CALL STACAP STEAI1800
UTIME(2)=DTSTAN/10.0 STEAI1810
FTIME(2)=UTIME(2)*20.9 STEAI1820
ITAPL(2)=7 STEAI1830

```

```

NITER=NT STEA1840
C PRINT THE STEADY STATE TEMPERATURE RESULTS BEFORE RETURNING STEA1850
C TO MP2. STEA1860
C CALL PRINT STEA1870
C SET AN INDICATOR TO SHOW THAT THE STEADY STATE ITERATIONS HAVE STEA1880
C BEEN COMPLETED. STEA1890
C CS1=2.0 STEA1900
C =====STEA1910
C RETURN TO MP2 AND PERFORM THE SMOOTHING ITERATIONS. STEA1920
C =====STEA1930
C RETURN STEA1940
C =====STEA1950
C SET THE INDICATOR IDT TO SHOW THE MODE OF DETERMINING THE NEXT STEA1960
C VALUE OF THE TIME STEP AND START THE NEXT SET OF ITERATIONS. STEA1970
C =====STEA1980
210 IDT=1 STEA1990
C GO TO 230 STEA2000
C IF INSTABILITIES OCCURRED IN SOLVING FOR TEMPERATURES,THE RETURN STEA2010
C FROM SUBROUTINE STEP WILL BE TO THE FOLLOWING STATEMENT. STEA2020
220 IDT=2 STEA2030
C IF (NSETT.NE.0) GO TO 230 STEA2040
C IDT=3 STEA2050
C GO TO 260 STEA2060
C =====STEA2070
C PRINT TEMPERATURES AND/OR RESIDUALS AT THE END OF THE PROBLEM. STEA2080
C =====STEA2090
C PREPARE THE ARRAY OF CONVERGENCE PARAMETERS AT THE END OF THE STEA2100
C PROBLEM (I.E.RESIDUALS)FOR PRINTING. STEA2110
300 DO 350 I=1,IMAX STEA2120
C DO 350 J=1,JMAX STEA2130
C DO 350 K=1,KMAX STEA2140
C IF (STEADY(I,J,K).GT.1.0E10) STEADY(I,J,K)=0.0 STEA2150
350 CONTINUE STEA2160
C PRINT THE LAST AVAILABLE TEMPERATURE RESULTS IF THE PROBLEM IS TO STEA2170
C TERMINATE IN THIS SUBROUTINE. STEA2180
C IF (NRES) GO TO 435 STEA2190
C NITER=NT STEA2200
C CALL TEMPS STEA2210
C PRINT THE RESIDUALS AT THE END OF THE PROBLEM. STEA2220
435 WRITE(6,440) STEA2230
440 FORMAT(1H0,56X,9HRESIDUALS) STEA2240
C CALL ARRAY(IDUM,STEADY,2) STEA2250
C IF (NRES) RETURN STEA2260
C PUNCH THE LAST AVAILABLE TEMPERATURE RESULTS IF THE PROBLEM IS TO STEA2270
C TERMINATE IN THIS SUBROUTINE. STEA2280
C CALL PUN STEA2290

```

```

C          CALL MERR                                     STEA2300
C          =====STE A2310
C          *****STE A2320
C          *****STE A2330
C          *****STE A2340
C          =====STE A2350
C          SUBROUTINE STACAP                             STEA2360
C          =====STE A2370
C          SET CAPACITANCE TERMS FOR MAXIMUM STABILITY. STEA2380
C          =====STE A2390
C          DO 60 I=2,IM                                  STEA2400
C          DO 60 J=2,JM                                  STEA2410
C          DO 60 K=2,KM                                  STEA2420
C          IF(RCP(I,J,K).GT.1.0E10) GO TO 80             STEA2430
C          RCP(I,J,K) = RR(I-1,J,K)*KR(I-1,J,K) + RR(I,J,K)*KR(I,J,K) +
C          .       RZ(I,J-1,K)*KZ(I,J-1,K) + RZ(I,J,K)*KZ(I,J,K) +
C          .       RT(I,J,K-1)*KT(I,J,K-1) + RT(I,J,K)*KT(I,J,K) STEA2440
C          CONTINUE                                     STEA2450
C          RETURN                                       STEA2460
C          =====STE A2470
C          *****STE A2480
C          *****STE A2490
C          *****STE A2500
C          *****STE A2510
C          *****STE A2520
C          =====STE A2530
C          SUBROUTINE STORE(ISTORE,NRW)                 STEA2540
C          =====STE A2550
C          WRITE ARRAYS ON DRUM OR READ ARRAYS FROM DRUM. STEA2560
C          =====STE A2570
C          REWIND ISTORE                                STEA2580
C          GO TO (100,200),NRW                          STEA2590
C          100 READ (ISTORE) (TI(N), N = 1,NQ), (TO(N), N = 1,NQ), STEA2600
C          X(((I(I,J,K), I=1,IQ), J=1,JQ), K=1,KQ), ((RBT(L(I,J,K), I=1,IQ), STEA2610
C          XJ=1,JQ), K=1,KQ), ((RBT(H(I,J,K), I=1,IQ), J=1,JQ), K=1,KQ), STEA2620
C          X(((Z(BTL(I,J,K), I=1,IQ), J=1,IQ), K=1,KQ), ((ZBT(H(I,J,K), STEA2630
C          XI=1,IQ), J=1,IQ), K=1,KQ), ((TBT(L(I,J,K), I=1,IQ), J=1,IQ), STEA2640
C          XK=1,IQ), ((TBT(H(I,J,K), I=1,IQ), J=1,IQ), K=1,IQ), STEA2650
C          GO TO 300                                     STEA2660
C          =====STE A2670
C          200 WRITE (ISTORE) (TI(N), N = 1,NQ), (TO(N), N = 1,NQ), STEA2680
C          X(((I(I,J,K), I=1,IQ), J=1,JQ), K=1,KQ), ((RBT(L(I,J,K), I=1,IQ), STEA2690
C          XJ=1,JQ), K=1,KQ), ((RBT(H(I,J,K), I=1,IQ), J=1,JQ), K=1,KQ), STEA2700
C          X(((Z(BTL(I,J,K), I=1,IQ), J=1,IQ), K=1,KQ), ((ZBT(H(I,J,K), STEA2710
C          XI=1,IQ), J=1,IQ), K=1,KQ), ((TBT(L(I,J,K), I=1,IQ), J=1,IQ), STEA2720
C          XK=1,IQ), ((TBT(H(I,J,K), I=1,IQ), J=1,IQ), K=1,IQ), STEA2730
C          300 RETURN                                    STEA2740
C          =====STE A2750
C          END                                           STEA2760

```

```

SUBROUTINE BLOCK                                     BLOC 10
INCLUDE      COMDIM                                 BLOC 20
=====
C EVALUATE THE MATERIAL PROPERTIES OF EACH BLOCK    BLOC 30
C =====
C TURN THE GAS CONDUCTIVITY SWITCH OFF             BLOC 40
GASE = .FALSE.                                     BLOC 50
C EVALUATE THE PROPERTIES OF ALL BLOCKS            BLOC 60
DO 20 L=1,LMAX ,                                   BLOC 70
NI= MB(L)                                          BLOC 80
C CALCULATE THE LIMITS OF BLOCK L                  BLOC 90
=====
C POINT LIMITS                                     BLOC 100
IH= IH(L)                                          BLOC 110
IL= IL(L)                                          BLOC 120
JH= JH(L)                                          BLOC 130
JL= JL(L)                                          BLOC 140
KH= KH(L)                                          BLOC 150
KL= KL(L)                                          BLOC 160
C BOUINDING GAPLINES                               BLOC 170
IGL= IGR(IL-1)                                     BLOC 180
IGH= IGR(IH)                                       BLOC 190
JGL= JGZ(JL-1)                                     BLOC 200
JGH= JGZ(JH)                                       BLOC 210
KGL= KGT(KL-1)                                     BLOC 220
KGH= KGT(KH)                                       BLOC 230
=====
C IS THIS BLOCK A COOLANT --YES,NO--              BLOC 240
IF (NI.LE.0) GO TO 10                              BLOC 250
C CALCULATE THE PROPERTIES OF THE MATERIAL IN BLOCK L.
CALL MADATA                                         BLOC 260
GO TO 20                                           BLOC 270
=====
C CORRECT THE GAPLINE ASSIGNMENT OF THE EXTERNAL COOLANTS
10 IF (IH.EQ.IMAX) IGH=IGL                          BLOC 280
IF (IL.EQ. 1) IGL=IGH                              BLOC 290
IF (JH.EQ. JMAX) JGH=JGL                          BLOC 300
IF (JL.EQ. 1) JGL=JGH                              BLOC 310
IF (KH.EQ. KMAX) KGH=KGL                          BLOC 320
IF (KL.EQ. 1) KGL=KGH                              BLOC 330
C CALCULATE THE PROPERTIES OF THE COOLANT IN BLOCK L
CALL FLUDAT(L)                                     BLOC 340
C ARE ALL BLOCKS EVALUATED --YES,NO--             BLOC 350
20 CONTINUE                                         BLOC 360

```

```

C =====BLOC 460
C ISOLATE THE CONDUCTIVITIES OF POINTS ADJACENT COOLANT BOUNDARIES. BLOC 470
C STORE THEM IN THE REM=RBBT, ZEM=ZBBT, TEM=TBTT LOCATIONS FOR BLOC 480
C COOLANT BLOCK SURFACES, BLOC 490
C =====BLOC 500
C LOCATE THE RADIAL-X COOLANT BOUNDARIES. BLOC 510
DO 220 I=1,IM BLOC 520
IG=IGR(1) BLOC 530
C EXCLUDE THE NON-GAPLINES. BLOC 540
IF(IG.EQ.0)GO TO 220 BLOC 550
DO 210 J=2,JM BLOC 560
DO 210 K=2,KM BLOC 570
C EXCLUDE THE POINTS WHERE A GAP IS ADJACENT TO THE GAPLINE. BLOC 580
IF(GAPR(IG,J,K).GE.0.0)GO TO 210 BLOC 590
C EXCLUDE THE POINTS WHERE TWO COOLANTS ARE ADJACENT ON THE GAPLINE. BLOC 600
IF(CAPR(IG,J,K).LT.(-2.5E-10))GO TO 210 BLOC 610
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE? -'NO,YES,- BLOC 620
IF(GAPR(IG,J,K).LT.(-1.5E-10))GO TO 200 BLOC 630
REML(IG,J,K)=CONR(I,J,K) BLOC 640
RBBTL(IG,J,K)=CONR(I+1,J,K) BLOC 650
GO TO 210 BLOC 660
200 REML(IG,J,K)=CONR(I+1,J,K) BLOC 670
RBBTL(IG,J,K)=CONR(I,J,K) BLOC 680
210 CONTINUE BLOC 690
220 CONTINUE BLOC 700
C =====BLOC 710
C LOCATE THE AXIAL-Z COOLANT BOUNDARIES. BLOC 720
DO 250 J=1,JM BLOC 730
JG=JGZ(J) BLOC 740
C EXCLUDE THE NON-GAPLINES. BLOC 750
IF(JG.EQ.0)GO TO 250 BLOC 760
DO 240 I=2,IM BLOC 770
DO 240 K=2,KM BLOC 780
C EXCLUDE THE POINTS WHERE A GAP IS ADJACENT TO THE GAPLINE. BLOC 790
IF(CAPZ(JG,I,K).GE.0.0)GO TO 240 BLOC 800
C EXCLUDE THE POINTS WHERE TWO COOLANTS ARE ADJACENT ON THE GAPLINE. BLOC 810
IF(CAPZ(JG,I,K).LT.(-2.5E-10))GO TO 240 BLOC 820
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE? -'NO,YES,- BLOC 830
IF(CAPZ(JG,I,K).LT.(-1.5E-10))GO TO 230 BLOC 840
ZEML(JG,I,K)=CONZ(I,J,K) BLOC 850
ZBBTL(JG,I,K)=CONZ(I,J+1,K) BLOC 860
GO TO 240 BLOC 870
230 ZEMH(JG,I,K)=CONZ(I,J+1,K) BLOC 880
ZBBH(JG,I,K)=CONZ(I,J,K) BLOC 890
240 CONTINUE BLOC 900
250 CONTINUE BLOC 910

```

```

C =====BLOC 920
C LOCATE THE THETA-Y COOLANT BOUNDARIES. BLOC 930
DO 300 K=1,KM BLOC 940
KG=NOT(K) BLOC 950
C EXCLUDE THE NON-GAPLINES. BLOC 960
IF (KG.EQ.0) GO TO 300 BLOC 970
DO 290 I=2,IM BLOC 980
DO 290 J=2,JM BLOC 990
C EXCLUDE THE POINTS WHERE A GAP IS ADJACENT TO THE GAPLINE. BLOC 1000
IF (GAPT(KG,I,J) .GE. 0.0) GO TO 290 BLOC 1010
C EXCLUDE THE POINTS WHERE TOO COOLANTS ARE ADJACENT ON THE GAPLINE. BLOC 1020
IF (GAPT(KG,I,J) .LT. (-2.5E-10))GO TO 290 BLOC 1030
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE? -'NO,YES,- BLOC 1040
IF (GAPT(KG,I,J) .LT. (-1.5E-10)) GO TO 280 BLOC 1050
TEMP(KG,I,J)=CONT(I,J,K) BLOC 1060
TBBIL(KG,I,J)=CONT(I,J,K+1) BLOC 1070
GO TO 290 BLOC 1080
280 TLMH(KG,I,J)=CONT(I,J,K+1) BLOC 1090
TBBIH(KG,I,J)=CONT(I,J,K) BLOC 1100
290 CONTINUE BLOC 1110
300 CONTINUE BLOC 1120
C =====BLOC 1130
RETURN BLOC 1140
END BLOC 1150

```


SUBROUTINE MAQATA	MADA 10
INCLUDE COMDIM	MADA 20
=====	MADA 30
C SUPPLY THE MATERIAL THERMAL PHYSICAL PROPERTIES	MADA 40
=====	MADA 50
=====	MADA 60
=====	MADA 70
=====	MADA 80
=====	MADA 90
=====	MADA 100
=====	MADA 110
=====	MADA 120
C IS THE MATERIAL SOLID OR GAS IN A GAP? -'GAS,SOLID'-	MADA 130
98 IF (GAS) GO TO 1300	MADA 140
=====	MADA 150
CB CALCULATE THE SOLID MATERIAL PROPERTIES	MADA 160
C/ CALCULATE THE HEATGENERATION RATE FOR EACH POINT IN THE BLOCK	MADA 170
=====	MADA 180
100 HR=DATI	MADA 190
DO 150 K = KLS, KHS	MADA 200
FTT = TP(K)	MADA 210
DO 150 J=JLS,JHS	MADA 220
FTZ = ZP(J)	MADA 230
DO 150 I=ILS,IHS	MADA 240
FTR=RP(I)	MADA 250
DK = T(I,J,K)	MADA 260
GO TO (101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,	MADA 270
1 112, 113, 114, 115),NI	MADA 280
101 X=HLAT 1(Y)	MADA 290
GO TO 140	MADA 300
102 X=HLAT 2(Y)	MADA 310
GO TO 140	MADA 320
103 X=HLAT 3(Y)	MADA 330
GO TO 140	MADA 340
104 X=HLAT 4(Y)	MADA 350
GO TO 140	MADA 360
105 X=HLAT 5(Y)	MADA 370
GO TO 140	MADA 380
106 X=HLAT 6(Y)	MADA 390
GO TO 140	MADA 400
107 X=HLAT 7(Y)	MADA 410
GO TO 140	MADA 420
108 X=HLAT 8(Y)	MADA 430
GO TO 140	MADA 440
109 X=HLAT 9(Y)	MADA 450

	GO TO 140	MADA 460
110	X=HLAT10(Y)	MADA 470
	GO TO 140	MADA 480
111	X=HLAT11(Y)	MADA 490
	GO TO 140	MADA 500
112	X=HLAT12(Y)	MADA 510
	GO TO 140	MADA 520
113	X=HLAT13(Y)	MADA 530
	GO TO 140	MADA 540
114	X=HLAT14(Y)	MADA 550
	GO TO 140	MADA 560
115	X=HLAT15(Y)	MADA 570
140	W(I,J,K)=X*V(I,J,K)	MADA 580
150	CONTINUE	MADA 590
CE	*****	MADA 600
C/	CALCULATE THE RADIAL CONDUCTIVITY	MADA 610
C=====		MADA 620
200	DO 250 K = KLS, KHS	MADA 630
	FTI=TP(K)	MADA 640
	DO 250 I=ILS, IHS	MADA 650
	FTR=RP(I)	MADA 660
	DO 250 J=JLS, JHS	MADA 670
	FTZ=ZP(J)	MADA 680
	DR = T(I,J,K)	MADA 690
	GO TO (201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211,	MADA 700
1	212, 213, 214, 215),NI	MADA 710
201	X=RCUN 1(Y)	MADA 720
	GO TO 247	MADA 730
202	X=RCUN 2(Y)	MADA 740
	GO TO 247	MADA 750
203	X=RCUN 3(Y)	MADA 760
	GO TO 247	MADA 770
204	X=RCUN 4(Y)	MADA 780
	GO TO 247	MADA 790
205	X=RCUN 5(Y)	MADA 800
	GO TO 247	MADA 810
206	X=RCUN 6(Y)	MADA 820
	GO TO 247	MADA 830
207	X=RCUN 7(Y)	MADA 840
	GO TO 247	MADA 850
208	X=RCUN 8(Y)	MADA 860
	GO TO 247	MADA 870
209	X=RCUN 9(Y)	MADA 880
	GO TO 247	MADA 890
210	X=RCUN10(Y)	MADA 900
	GO TO 247	MADA 910

211	X=RCON11(Y)	MADA 920
	GO TO 247	MADA 930
212	X=RCON12(Y)	MADA 940
	GO TO 247	MADA 950
213	X=RCON13(Y)	MADA 960
	GO TO 247	MADA 970
214	X=RCON14(Y)	MADA 980
	GO TO 247	MADA 990
215	X=RCON15(Y)	MADA1000
247	IF(A.EQ.0.0)X=1.0E-6	MADA1010
	CONK(I,J,K)=1.0/X	MADA1020
250	CONTINUE	MADA1030
CE	*****	MADA1040
C/	CALCULATE THE AXIAL CONDUCTIVITY	MADA1050
C=====		MADA1060
300	DO 350 K = KLS, KHS	MADA1070
	FTI=TP(K)	MADA1080
	DO 350 I=ILS, IHS	MADA1090
	FTI=RP(I)	MADA1100
	DO 350 J=JLS, JHS	MADA1110
	FTI=ZP(J)	MADA1120
	DK = T(I,J,K)	MADA1130
	GO TO (301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311,	MADA1140
1	312, 313, 314, 315),NI	MADA1150
301	X=ACON 1(Y)	MADA1160
	GO TO 347	MADA1170
302	X=ACON 2(Y)	MADA1180
	GO TO 347	MADA1190
303	X=ACON 3(Y)	MADA1200
	GO TO 347	MADA1210
304	X=ACON 4(Y)	MADA1220
	GO TO 347	MADA1230
305	X=ACON 5(Y)	MADA1240
	GO TO 347	MADA1250
306	X=ACON 6(Y)	MADA1260
	GO TO 347	MADA1270
307	X=ACON 7(Y)	MADA1280
	GO TO 347	MADA1290
308	X=ACON 8(Y)	MADA1300
	GO TO 347	MADA1310
309	X=ACON 9(Y)	MADA1320
	GO TO 347	MADA1330
310	X=ACON10(Y)	MADA1340
	GO TO 347	MADA1350
311	X=ACON11(Y)	MADA1360
	GO TO 347	MADA1370

312	X=ACON12(Y)	MADA1380
	GO TO 347	MADA1390
313	X=ACON13(Y)	MADA1400
	GO TO 347	MADA1410
314	X=ACON14(Y)	MADA1420
	GO TO 347	MADA1430
315	X=ACON15(Y)	MADA1440
347	IF (A.EQ.0.0)X=1.0E-6	MADA1450
	CONL(I,J,K)=1.0/X	MADA1460
350	CONTINUE	MADA1470
CE	*****	MADA1480
C/	CALCULATE THE THETA CONDUCTIVITY	MADA1490
C=====	=====	MADA1500
1500	DO 1550 K = KLS, KHS	MADA1510
	FTT=TP(K)	MADA1520
	DO 1550 I = ILS, IHS	MADA1530
	FTR=RP(I)	MADA1540
	DO 1550 J = JLS, JHS	MADA1550
	FTZ=ZP(J)	MADA1560
	OK = T(I,J,K)	MADA1570
	GO TO (1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1510,	MADA1580
	1511, 1512, 1513, 1514, 1515), NI	MADA1590
1501	X=TCON 1(Y)	MADA1600
	GO TO 1547	MADA1610
1502	X=TCON 2(Y)	MADA1620
	GO TO 1547	MADA1630
1503	X=TCON 3(Y)	MADA1640
	GO TO 1547	MADA1650
1504	X=TCON 4(Y)	MADA1660
	GO TO 1547	MADA1670
1505	X=TCON 5(Y)	MADA1680
	GO TO 1547	MADA1690
1506	X=TCON 6(Y)	MADA1700
	GO TO 1547	MADA1710
1507	X=TCON 7(Y)	MADA1720
	GO TO 1547	MADA1730
1508	X=TCON 8(Y)	MADA1740
	GO TO 1547	MADA1750
1509	X=TCON 9(Y)	MADA1760
	GO TO 1547	MADA1770
1510	X=TCON10(Y)	MADA1780
	GO TO 1547	MADA1790
1511	X=TCON11(Y)	MADA1800
	GO TO 1547	MADA1810
1512	X=TCON12(Y)	MADA1820
	GO TO 1547	MADA1830

```

1513 X=TCGN13(Y)                                MADA1840
      GO TO 1547                                MADA1850
1514 X=TCGN14(Y)                                MADA1860
      GO TO 1547                                MADA1870
1515 X=TCGN15(Y)                                MADA1880
1547 IF (X.EQ.0.0)X=1.0E-6                      MADA1890
      CON(I,J,K)=1.0/X                          MADA1900
1550 CONTINUE                                    MADA1910
CE *****MADA1920
C/  CALCULATE THE SPECIFIC HEAT                 MADA1930
C=====
      IF (CG2.GT.1.0) GO TO 499                 MADA1940
400  DO 450 K = KLS, KHS                        MADA1950
      DO 450 I=ILS, IHS                        MADA1960
      DO 450 J=JLS, JHS                        MADA1970
      DR = 1(I,J,K)                            MADA1980
      GO TO ( 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411,
1      412, 413, 414, 415),NI                 MADA1990
401  X=SPEC 1(Y)                                MADA2000
      GO TO 440                                MADA2010
402  X=SPEC 2(Y)                                MADA2020
      GO TO 440                                MADA2030
403  X=SPEC 3(Y)                                MADA2040
      GO TO 440                                MADA2050
404  X=SPEC 4(Y)                                MADA2060
      GO TO 440                                MADA2070
405  X=SPEC 5(Y)                                MADA2080
      GO TO 440                                MADA2090
406  X=SPEC 6(Y)                                MADA2100
      GO TO 440                                MADA2110
407  X=SPEC 7(Y)                                MADA2120
      GO TO 440                                MADA2130
408  X=SPEC 8(Y)                                MADA2140
      GO TO 440                                MADA2150
409  X=SPEC 9(Y)                                MADA2160
      GO TO 440                                MADA2170
410  X=SPEC10(Y)                               MADA2180
      GO TO 440                                MADA2190
411  X=SPEC11(Y)                               MADA2200
      GO TO 440                                MADA2210
412  X=SPEC12(Y)                               MADA2220
      GO TO 440                                MADA2230
413  X=SPEC13(Y)                               MADA2240
      GO TO 440                                MADA2250
414  X=SPEC14(Y)                               MADA2260
      GO TO 440                                MADA2270
      GO TO 440                                MADA2280
      GO TO 440                                MADA2290

```

415 X=SPEC15(Y)	MADA2300
440 RCP(I,J,K)=X*V(I,J,K)	MADA2310
IF(.NOT.SW(10)) GO TO 450	MADA2320
IF(X.GT.9.999E5) RCP(I,J,K)=1.0E12	MADA2330
450 CONTINUE	MADA2340
CE *****	MADA2350
C/ CALCULATE THE RADIAL THERMAL EMISSIVITY ON THE LOW INDEX	MADA2360
C .SIDE OF THE BLOCK	MADA2370
C=====	MADA2380
499 IF (IGLS . LE . 0) GO TO 599	MADA2390
500 DO 550 K = KLS, KHS	MADA2400
DO 550 J=JLS,JHS	MADA2410
IF (GAPR(IGLS,J,K) . EQ . 0.0) GO TO 550	MADA2420
FTZ=ZP(K)	MADA2430
FTZ=ZP(J)	MADA2440
UR=URBTL(IGLS,J,K)	MADA2450
GO TO (501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511,	MADA2460
1 512, 513, 514, 515),NI	MADA2470
501 X=EMRL 1(Y)	MADA2480
GO TO 547	MADA2490
502 X=EMRL 2(Y)	MADA2500
GO TO 547	MADA2510
503 X=EMRL 3(Y)	MADA2520
GO TO 547	MADA2530
504 X=EMRL 4(Y)	MADA2540
GO TO 547	MADA2550
505 X=EMRL 5(Y)	MADA2560
GO TO 547	MADA2570
506 X=EMRL 6(Y)	MADA2580
GO TO 547	MADA2590
507 X=EMRL 7(Y)	MADA2600
GO TO 547	MADA2610
508 X=EMRL 8(Y)	MADA2620
GO TO 547	MADA2630
509 X=EMRL 9(Y)	MADA2640
GO TO 547	MADA2650
510 X=EMRL10(Y)	MADA2660
GO TO 547	MADA2670
511 X=EMRL11(Y)	MADA2680
GO TO 547	MADA2690
512 X=EMRL12(Y)	MADA2700
GO TO 547	MADA2710
513 X=EMRL13(Y)	MADA2720
GO TO 547	MADA2730
514 X=EMRL14(Y)	MADA2740
GO TO 547	MADA2750

515	X=EMRL15(Y)	MADA2760
547	IF(X.EQ.0.0)X=1.0E-6	MADA2770
	REML(IGLS,J,K)=1.0/X	MADA2780
550	CONTINUE	MADA2790
CE	*****	MADA2800
C/	CALCULATE THE RADIAL THERMAL EMISSIVITY ON THE HIGH INDEX	MADA2810
C	SIDE OF THE BLOCK	MADA2820
	=====	MADA2830
599	IF (IGHS . LE . 0) GO TO 699	MADA2840
600	DO 650 K = KLS, KHS	MADA2850
	DO 650 J=JLS, JHS	MADA2860
	IF (GAPK(IGHS,J,K) . EQ . 0.0) GO TO 650	MADA2870
	FTT=TP(K)	MADA2880
	FTZ=ZP(J)	MADA2890
	DK=KBDTH(IGHS,J,K)	MADA2900
	GO TO (601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611,	MADA2910
	1 612, 613, 614, 615),NI	MADA2920
601	X=EMRH 1(Y)	MADA2930
	GO TO 647	MADA2940
602	X=EMRH 2(Y)	MADA2950
	GO TO 647	MADA2960
603	X=EMRH 3(Y)	MADA2970
	GO TO 647	MADA2980
604	X=EMRH 4(Y)	MADA2990
	GO TO 647	MADA3000
605	X=EMRH 5(Y)	MADA3010
	GO TO 647	MADA3020
606	X=EMRH 6(Y)	MADA3030
	GO TO 647	MADA3040
607	X=EMRH 7(Y)	MADA3050
	GO TO 647	MADA3060
608	X=EMRH 8(Y)	MADA3070
	GO TO 647	MADA3080
609	X=EMRH 9(Y)	MADA3090
	GO TO 647	MADA3100
610	X=EMRH10(Y)	MADA3110
	GO TO 647	MADA3120
611	X=EMRH11(Y)	MADA3130
	GO TO 647	MADA3140
612	X=EMRH12(Y)	MADA3150
	GO TO 647	MADA3160
613	X=EMRH13(Y)	MADA3170
	GO TO 647	MADA3180
614	X=EMRH14(Y)	MADA3190
	GO TO 647	MADA3200
615	X=EMRH15(Y)	MADA3210

647 IF (X.EQ.0.0)X=1.0E-6	MADA3220
REMP(I,GS,J,K)=1.0/X	MADA3230
650 CONTINUE	MADA3240
CE *****	MADA3250
C/ CALCULATE THE AXIAL THERMAL EMISSIVITY ON THE LOW INDEX	MADA3260
C SIDE OF THE BLOCK	MADA3270
C=====	MADA3280
699 IF (JGLS . LE . 0) GO TO 799	MADA3290
700 DO 750 K = KLS, KHS	MADA3300
DO 750 I = ILS, IHS	MADA3310
IF (GAP/(JGLS,I,K) . EQ . 0.0) GO TO 750	MADA3320
FTR=RP(K)	MADA3330
FTR=RP(I)	MADA3340
JR = ZIBTL(JGLS,I,K)	MADA3350
GO TO (701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711,	MADA3360
1 712, 713, 714, 715),NI	MADA3370
701 X=EMAL 1(Y)	MADA3380
GO TO 747	MADA3390
702 X=EMAL 2(Y)	MADA3400
GO TO 747	MADA3410
703 X=EMAL 3(Y)	MADA3420
GO TO 747	MADA3430
704 X=EMAL 4(Y)	MADA3440
GO TO 747	MADA3450
705 X=EMAL 5(Y)	MADA3460
GO TO 747	MADA3470
706 X=EMAL 6(Y)	MADA3480
GO TO 747	MADA3490
707 X=EMAL 7(Y)	MADA3500
GO TO 747	MADA3510
708 X=EMAL 8(Y)	MADA3520
GO TO 747	MADA3530
709 X=EMAL 9(Y)	MADA3540
GO TO 747	MADA3550
710 X=EMAL10(Y)	MADA3560
GO TO 747	MADA3570
711 X=EMAL11(Y)	MADA3580
GO TO 747	MADA3590
712 X=EMAL12(Y)	MADA3600
GO TO 747	MADA3610
713 X=EMAL13(Y)	MADA3620
GO TO 747	MADA3630
714 X=EMAL14(Y)	MADA3640
GO TO 747	MADA3650
715 X=EMAL15(Y)	MADA3660
747 IF (X.EQ.0.0)X=1.0E-6	MADA3670


```

      ZEML(JGLS,I,K)=1.0/X
750 CONTINUE
CE *****
C/ CALCULATE THE AXIAL THERMAL EMISSIVITY ON THE HIGH INDEX
C  .SIDL OF THE BLOCK
C=====
799 IF (JGHS . LE . 0 ) GO TO 899
800 DO 050 K = KLS, KHS
      DO 050 I = ILS, IHS
        IF (GAPZ(JGHS, I,K) . EQ . 0,0) GO TO 850
        FTT=TP(K)
        FIR=RP(I)
        DR = ZBOTH (JGHS, I,K)
        GO TO ( 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811,
1         812, 813, 814, 815),NI
801 X=EMAH 1(Y)
      GO TO 847
802 X=EMAH 2(Y)
      GO TO 847
803 X=EMAH 3(Y)
      GO TO 847
804 X=EMAH 4(Y)
      GO TO 847
805 X=EMAH 5(Y)
      GO TO 847
806 X=EMAH 6(Y)
      GO TO 847
807 X=EMAH 7(Y)
      GO TO 847
808 X=EMAH 8(Y)
      GO TO 847
809 X=EMAH 9(Y)
      GO TO 847
810 X=EMAH10(Y)
      GO TO 847
811 X=EMAH11(Y)
      GO TO 847
812 X=EMAH12(Y)
      GO TO 847
813 X=EMAH13(Y)
      GO TO 847
814 X=EMAH14(Y)
      GO TO 847
815 X=EMAH15(Y)
847 IF (X.EQ.0.0)X=1.0E-6
      ZEMH(JGHS,I,K)=1.0/X
MADA3680
MADA3690
MADA3700
MADA3710
MADA3720
MADA3730
MADA3740
MADA3750
MADA3760
MADA3770
MADA3780
MADA3790
MADA3800
MADA3810
MADA3820
MADA3830
MADA3840
MADA3850
MADA3860
MADA3870
MADA3880
MADA3890
MADA3900
MADA3910
MADA3920
MADA3930
MADA3940
MADA3950
MADA3960
MADA3970
MADA3980
MADA3990
MADA4000
MADA4010
MADA4020
MADA4030
MADA4040
MADA4050
MADA4060
MADA4070
MADA4080
MADA4090
MADA4100
MADA4110
MADA4120
MADA4130

```

850	CONTINUE	MADA4140
CE	*****	MADA4150
C/	CALCULATE THE THETA THERMAL EMISSIVITY ON THE LOW INDEX	MADA4160
C	.SIDE OF THE BLOCK	MADA4170
	=====	MADA4180
899	IF(KGLS . LE . 0)GO TO 999	MADA4190
900	DO 950 I = ILS, IHS	MADA4200
	DO 950 J = JLS, JHS	MADA4210
	IF (GAPT(KGLS,I,J) . EQ . 0.0) GO TO 950	MADA4220
	FTR=RP(I)	MADA4230
	FTZ=P(J)	MADA4240
	OR = INTL(KGLS,I,J)	MADA4250
	GO TO (901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911,	MADA4260
	912, 913, 914, 915), NI	MADA4270
901	X=EMTL 1(Y)	MADA4280
	GO TO 947	MADA4290
902	X=EMTL 2(Y)	MADA4300
	GO TO 947	MADA4310
903	X=EMTL 3(Y)	MADA4320
	GO TO 947	MADA4330
904	X=EMTL 4(Y)	MADA4340
	GO TO 947	MADA4350
905	X=EMTL 5(Y)	MADA4360
	GO TO 947	MADA4370
906	X=EMTL 6(Y)	MADA4380
	GO TO 947	MADA4390
907	X=EMTL 7(Y)	MADA4400
	GO TO 947	MADA4410
908	X=EMTL 8(Y)	MADA4420
	GO TO 947	MADA4430
909	X=EMTL 9(Y)	MADA4440
	GO TO 947	MADA4450
910	X=EMTL10(Y)	MADA4460
	GO TO 947	MADA4470
911	X=EMTL11(Y)	MADA4480
	GO TO 947	MADA4490
912	X=EMTL12(Y)	MADA4500
	GO TO 947	MADA4510
913	X=EMTL13(Y)	MADA4520
	GO TO 947	MADA4530
914	X=EMTL14(Y)	MADA4540
	GO TO 947	MADA4550
915	X=EMTL15(Y)	MADA4560
947	IF (X.EQ.0.0)X=1.0E-6	MADA4570
	TLML(KGLS,I,J)=1.0/X	MADA4580
950	CONTINUE	MADA4590

```

CE *****MADA4600
C/ CALCULATE THE THETA THERMAL EMISSIVITY ON THE HIGH INDEX MADA4610
C .SIDE OF THE BLOCK MADA4620
C=====MADA4630
999 IF (KGHS . LE . 0) GO TO 1299 MADA4640
1000 DO 1050 I = ILS, IHS MADA4650
      DO 1050 J = JLS, JHS MADA4660
      IF (GAPT(KGHS,I,J) . EQ . 0,0) GO TO 1299 MADA4670
      FTR=RP(I) MADA4680
      FTZ=ZP(J) MADA4690
      DR = DRTH(KGHS,I,J) MADA4700
      GO TO (1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009,
1001, 1010, 1011, 1012, 1013, 1014, 1015), NI MADA4710
1001 X=EMTH 1(Y) MADA4720
      GO TO 1047 MADA4730
1002 X=EMTH 2(Y) MADA4740
      GO TO 1047 MADA4750
1003 X=EMTH 3(Y) MADA4760
      GO TO 1047 MADA4770
1004 X=EMTH 4(Y) MADA4780
      GO TO 1047 MADA4790
1005 X=EMTH 5(Y) MADA4800
      GO TO 1047 MADA4810
1006 X=EMTH 6(Y) MADA4820
      GO TO 1047 MADA4830
1007 X=EMTH 7(Y) MADA4840
      GO TO 1047 MADA4850
1008 X=EMTH 8(Y) MADA4860
      GO TO 1047 MADA4870
1009 X=EMTH 9(Y) MADA4880
      GO TO 1047 MADA4890
1010 X=EMTH10(Y) MADA4900
      GO TO 1047 MADA4910
1011 X=EMTH11(Y) MADA4920
      GO TO 1047 MADA4930
1012 X=EMTH12(Y) MADA4940
      GO TO 1047 MADA4950
1013 X=EMTH13(Y) MADA4960
      GO TO 1047 MADA4970
1014 X=EMTH14(Y) MADA4980
      GO TO 1047 MADA4990
1015 X=EMTH15(Y) MADA5000
1047 IF (X.EQ.0.0)X=1.0E-6 MADA5010
      TLMH(KGHS,I,J)=1.0/X MADA5020
1050 CONTINUE MADA5030
CE *****MADA5040
*****MADA5050

```

```

CE *****MADA5060
1299 RETURN MADA5070
C=====MADA5080
C/ CALCULATE THE CONDUCTIVITY OF THE GAS IN THE GAP MADA5090
C=====MADA5100
1300 GK=(TH+TB)*0.5 MADA5110
      GO TO (1301,1302,1303,1304,1305,1306,1307,1308,1309,1310,1311,
      1 1312,1313,1314,1315),NI MADA5120
1301 GK=CCON1(X) MADA5130
      GO TO 1350 MADA5140
1302 GK=CCON2(X) MADA5150
      GO TO 1350 MADA5160
1303 GK=CCON3(X) MADA5170
      GO TO 1350 MADA5180
1304 GK=CCON4(X) MADA5190
      GO TO 1350 MADA5200
1305 GK=CCON5(X) MADA5210
      GO TO 1350 MADA5220
1306 GK=CCON6(X) MADA5230
      GO TO 1350 MADA5240
1307 GK=CCON7(X) MADA5250
      GO TO 1350 MADA5260
1308 GK=CCON8(X) MADA5270
      GO TO 1350 MADA5280
1309 GK=CCON9(X) MADA5290
      GO TO 1350 MADA5300
1310 GK=CCON10(X) MADA5310
      GO TO 1350 MADA5320
1311 GK=CCON11(X) MADA5330
      GO TO 1350 MADA5340
1312 GK=CCON12(X) MADA5350
      GO TO 1350 MADA5360
1313 GK=CCON13(X) MADA5370
      GO TO 1350 MADA5380
1314 GK=CCON14(X) MADA5390
      GO TO 1350 MADA5400
1315 GK=CCON15(X) MADA5410
1350 CONTINUE MADA5420
CE *****MADA5430
      RETURN MADA5440
      END MADA5450
      MADA5460

```

```

FUNCTION FMAT1 (X)
ENTRY HEAT 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY RCON 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY ACON 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY TCON 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY SPEC 1(X)
FMAT1 = 1.0
GO TO 10
ENTRY EMRL 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY EMRH 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY EMAL 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY EMAM 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY EMTL 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY EMTH 1(X)
FMAT1 = 0.0
GO TO 10
ENTRY GCON 1(X)
FMAT1 = 0.0
10 RETURN
END

```

```

FMAT 10
FMAT 20
FMAT 30
FMAT 40
FMAT 50
FMAT 60
FMAT 70
FMAT 80
FMAT 90
FMAT 100
FMAT 110
FMAT 120
FMAT 130
FMAT 140
FMAT 150
FMAT 160
FMAT 170
FMAT 180
FMAT 190
FMAT 200
FMAT 210
FMAT 220
FMAT 230
FMAT 240
FMAT 250
FMAT 260
FMAT 270
FMAT 280
FMAT 290
FMAT 300
FMAT 310
FMAT 320
FMAT 330
FMAT 340
FMAT 350
FMAT 360
FMAT 370
FMAT 380

```

FUNCTION FMAT2 (X)	FMAT 10
ENTRY HEAT 2(X)	FMAT 20
FMAT2 = 0.0	FMAT 30
GO TO 10	FMAT 40
ENTRY RCON 2(X)	FMAT 50
FMAT2 = 0.0	FMAT 60
GO TO 10	FMAT 70
ENTRY ACON 2(X)	FMAT 80
FMAT2 = 0.0	FMAT 90
GO TO 10	FMAT 100
ENTRY TCON 2(X)	FMAT 110
FMAT2 = 0.0	FMAT 120
GO TO 10	FMAT 130
ENTRY SPEC 2(X)	FMAT 140
FMAT2 = 1.0	FMAT 150
GO TO 10	FMAT 160
ENTRY EMRL 2(X)	FMAT 170
FMAT2 = 0.0	FMAT 180
GO TO 10	FMAT 190
ENTRY EMRH 2(X)	FMAT 200
FMAT2 = 0.0	FMAT 210
GO TO 10	FMAT 220
ENTRY EMAL 2(X)	FMAT 230
FMAT2 = 0.0	FMAT 240
GO TO 10	FMAT 250
ENTRY EMAM 2(X)	FMAT 260
FMAT2 = 0.0	FMAT 270
GO TO 10	FMAT 280
ENTRY EMTL 2(X)	FMAT 290
FMAT2 = 0.0	FMAT 300
GO TO 10	FMAT 310
ENTRY EMTH 2(X)	FMAT 320
FMAT2 = 0.0	FMAT 330
GO TO 10	FMAT 340
ENTRY GCON 2(X)	FMAT 350
FMAT2 = 0.0	FMAT 360
10 RETURN	FMAT 370
END	FMAT 380

NOTE:

The functions FMAT3 through FMAT15 are omitted since they are all of the form illustrated in the preceding listings of FMAT1 and FMAT2.

```

SUBROUTINE FLODAT (L)
INCLUDE COMDIM
=====
C SUPPLY THE COOLANT THERMAL PHYSICAL PROPERTIES
=====
C DIMENSION IFLODA( 4),STA ( 2)
DATA (IFLODA(I),I=1,4) /6HFLODAT, 1H1, 1H2, 1H3 /
INTEGER SELECT, TYP1, TYP2, TYP3
=====
C ERROR STOPS=
C FLODAT1 THE INDEPENDENT VALUE LIES OUTSIDE THE FLOW RATE
C FUNCTION RANGES.
C FLODAT2 THE INDEPENDENT VALUE LIES OUTSIDE THE INLET
C TEMPERATURE FUNCTION RANGES.
C FLODAT3 THE REYNOLDS NUMBER LIES OUTSIDE THE SPECIFIED RANGES
=====
C SUPPLY THE FLOW PROPERTIES
=====
C COOLANT NUMBER STORED AS NEGATIVE MATERIAL NUMBER
1 NEIAPS(MB(L))
=====
C IS THIS COOLANT IN USE? -'NO,YES,-
IF (FLOW(N).EQ.0.0) RETURN
=====
C ASSIGN THE CURRENT TIME, INLET AND OUTLET TEMPERATURE
HR = DATI
TIN=T1(N)
TOUT=T0(N)
=====
C/ DETERMINE THE RANGE OF THE FLOW FUNCTION TO BE USED
C .-'LOW,ERROR,MIDDLE,HIGH'-
IP=IFLO(N)
GO TO (600,52,53,54),IP
52 X=HR
GO TO 55
53 X=TOUT
GO TO 55
54 X=TIN
=====
C DECIDE WHICH FLOW FUNCTION TO USE
55 IF (X.LT.FLIM1(N)) GO TO 56
IF ( X . LT . FLIM2(N)) GO TO 600
=====

```

```

          IF (X . LT . FLIM3(N)) GO TO 700                                FLOD 460
          IF (X . LE . FLIM4(N)) GO TO 800                                FLOD 470
CE *****                                                                    FLOD 480
C THE INDEPENDENT VALUE LIES OUTSIDE ALL RANGES                          FLOD 490
  56 IERROR(1)=IFLODA(1)                                                FLOD 500
     IERROR(2)=IFLODA(2)                                                FLOD 510
     CALL ERROR?                                                         FLOD 520
C=====                                                                    FLOD 530
C/ EVALUATE THE FLOWRATE USING THE LOW RANGE FLOW FUNCTION .           FLOD 540
  600 GO TO(61,62,63,64,65,66,67,68,69,70,711,712,713,714,715),N       FLOD 550
C=====                                                                    FLOD 560
C LOWER RANGE FLOW RATE FUNCTION                                        FLOD 570
  61 FLOW(1) = FLO1A(X)                                                  FLOD 580
     GO TO 900                                                            FLOD 590
  62 FLOW(2) = FLO2A(X)                                                  FLOD 600
     GO TO 900                                                            FLOD 610
  63 FLOW(3) = FLO3A(X)                                                  FLOD 620
     GO TO 900                                                            FLOD 630
  64 FLOW(4) = FLO4A(X)                                                  FLOD 640
     GO TO 900                                                            FLOD 650
  65 FLOW(5) = FLO5A(X)                                                  FLOD 660
     GO TO 900                                                            FLOD 670
  66 FLOW(6) = FLO6A(X)                                                  FLOD 680
     GO TO 900                                                            FLOD 690
  67 FLOW(7) = FLO7A(X)                                                  FLOD 700
     GO TO 900                                                            FLOD 710
  68 FLOW(8) = FLO8A(X)                                                  FLOD 720
     GO TO 900                                                            FLOD 730
  69 FLOW(9) = FLO9A(X)                                                  FLOD 740
     GO TO 900                                                            FLOD 750
  70 FLOW(10) = FLO10A(X)                                                FLOD 760
     GO TO 900                                                            FLOD 770
  711 FLOW(11)=FLO11A(X)                                                 FLOD 780
     GO TO 900                                                            FLOD 790
  712 FLOW(12)=FLO12A(X)                                                 FLOD 800
     GO TO 900                                                            FLOD 810
  713 FLOW(13)=FLO13A(X)                                                 FLOD 820
     GO TO 900                                                            FLOD 830
  714 FLOW(14)=FLO14A(X)                                                 FLOD 840
     GO TO 900                                                            FLOD 850
  715 FLOW(15)=FLO15A(X)                                                 FLOD 860
     GO TO 900                                                            FLOD 870
CE *****                                                                    FLOD 880
C/ EVALUATE THE FLOWRATE USING THE MIDDLE RANGE FLOW FUNCTION          FLOD 890
  700 GO TO(71,72,73,74,75,76,77,78,79,80,811,812,813,814,815),N     FLOD 900
  71 FLOW(1) = FLO1B(X)                                                 FLOD 910

```


GO TO 900	FLOD 920
72 FLOW(2) = FLO2B(X)	FLOD 930
GO TO 900	FLOD 940
73 FLOW(3) = FLO3B(X)	FLOD 950
GO TO 900	FLOD 960
74 FLOW(4) = FLO4B(X)	FLOD 970
GO TO 900	FLOD 980
75 FLOW(5) = FLO5B(X)	FLOD 990
GO TO 900	FLOD1000
76 FLOW(6) = FLO6B(X)	FLOD1010
GO TO 900	FLOD1020
77 FLOW(7) = FLO7B(X)	FLOD1030
GO TO 900	FLOD1040
78 FLOW(8) = FLO8B(X)	FLOD1050
GO TO 900	FLOD1060
79 FLOW(9) = FLO9B(X)	FLOD1070
GO TO 900	FLOD1080
80 FLOW(10) = FLO10B(X)	FLOD1090
GO TO 900	FLOD1100
811 FLOW(11)=FLO11B(X)	FLOD1110
GO TO 900	FLOD1120
812 FLOW(12)=FLO12B(X)	FLOD1130
GO TO 900	FLOD1140
813 FLOW(13)=FLO13B(X)	FLOD1150
GO TO 900	FLOD1160
814 FLOW(14)=FLO14B(X)	FLOD1170
GO TO 900	FLOD1180
815 FLOW(15)=FLO15B(X)	FLOD1190
GO TO 900	FLOD1200
CE *****	FLOD1210
C/ EVALUATE THE FLOWRATE USING THE HIGH RANGE FLOW FUNCTION	FLOD1220
800 GO TO(81,82,83,84,85,86,87,88,89,90,911,912,913,914,915),N	FLOD1230
81 FLOW(1) = FLO1C(X)	FLOD1240
GO TO 900	FLOD1250
82 FLOW(2) = FLO2C(X)	FLOD1260
GO TO 900	FLOD1270
83 FLOW(3) = FLO3C(X)	FLOD1280
GO TO 900	FLOD1290
84 FLOW(4) = FLO4C(X)	FLOD1300
GO TO 900	FLOD1310
85 FLOW(5) = FLO5C(X)	FLOD1320
GO TO 900	FLOD1330
86 FLOW(6) = FLO6C(X)	FLOD1340
GO TO 900	FLOD1350
87 FLOW(7) = FLO7C(X)	FLOD1360
GO TO 900	FLOD1370

```

88 FLOW(8) = FLO8C(X)                                FLOD1380
GO TO 900                                            FLOD1390
89 FLOW(9) = FLO9C(X)                                FLOD1400
GO TO 900                                            FLOD1410
90 FLOW(10) = FLO10C(X)                              FLOD1420
GO TO 900                                            FLOD1430
911 FLOW(11)=FLO11C(X)                               FLOD1440
GO TO 900                                            FLOD1450
912 FLOW(12)=FLO12C(X)                               FLOD1460
GO TO 900                                            FLOD1470
913 FLOW(13)=FLO13C(X)                               FLOD1480
GO TO 900                                            FLOD1490
914 FLOW(14)=FLO14C(X)                               FLOD1500
GO TO 900                                            FLOD1510
915 FLOW(15)=FLO15C(X)                               FLOD1520
GO TO 900                                            FLOD1530
CE *****                                            FLOD1540
C DO NOT ALLOW A ZERO FLOWRATE                       FLOD1550
900 IF (FLOW(N).EQ.0.0) FLOW(N)=1.0E-6              FLOD1560
FR=FLOW(N)                                          FLOD1570
C=====FLOD1580
C/ DETERMINE THE RANGE OF THE INLET TEMPERATURE FUNCTION TO BE USED FLOD1590
C .-LOW,ERROR,MIDDLE,HIGH.-                         FLOD1600
IT=ITIN(N)                                         FLOD1610
100 GO TO (200,11,12,13), IT                       FLOD1620
11 X = HR                                          FLOD1630
GO TO 14                                           FLOD1640
12 X = FR                                          FLOD1650
GO TO 14                                           FLOD1660
13 X = TOUT                                        FLOD1670
GO TO 14                                           FLOD1680
C=====FLOD1690
14 IF (X.LT.TLIM1(N)) GO TO 15                     FLOD1700
IF (X . LT . TLIM2(N)) GO TO 200                  FLOD1710
IF (X . LT . TLIM3(N)) GO TO 300                  FLOD1720
IF ( X . LE . TLIM4(N)) GO TO 400                  FLOD1730
CE *****FLOD1740
C THE INDEPENDENT VALUE LIES OUTSIDE ALL RANGES    FLOD1750
15 IERROR(1)=IFLODA(1)                             FLOD1760
IERROR(2)=IFLODA(3)                                FLOD1770
CALL ERROR2                                         FLOD1780
C=====FLOD1790
C/ EVALUATE THE INLET TEMPERATURE USING THE LOW RANGE FUNCTION FLOD1800
200 GO TO(21,22,23,24,25,26,27,28,29,30,311,312,313,314,315),N FLOD1810
21 T1(1) = TIN1A(X)                                FLOD1820
GO TO 500                                          FLOD1830

```

22	TI(2) = TIN2A(X)	FLOD1840
	GO TO 500	FLOD1850
23	TI(3) = TIN3A(X)	FLOD1860
	GO TO 500	FLOD1870
24	TI(4) = TIN4A(X)	FLOD1880
	GO TO 500	FLOD1890
25	TI(5) = TIN5A(X)	FLOD1900
	GO TO 500	FLOD1910
26	TI(6) = TIN6A(X)	FLOD1920
	GO TO 500	FLOD1930
27	TI(7) = TIN7A(X)	FLOD1940
	GO TO 500	FLOD1950
28	TI(8) = TIN8A(X)	FLOD1960
	GO TO 500	FLOD1970
29	TI(9) = TIN9A(X)	FLOD1980
	GO TO 500	FLOD1990
30	TI(10) = TIN10A(X)	FLOD2000
	GO TO 500	FLOD2010
311	TI(11) = TIN11A(X)	FLOD2020
	GO TO 500	FLOD2030
312	TI(12) = TIN12A(X)	FLOD2040
	GO TO 500	FLOD2050
313	TI(13) = TIN13A(X)	FLOD2060
	GO TO 500	FLOD2070
314	TI(14) = TIN14A(X)	FLOD2080
	GO TO 500	FLOD2090
315	TI(15) = TIN15A(X)	FLOD2100
	GO TO 500	FLOD2110
CE	*****	FLOD2120
C/	EVALUATE THE INLET TEMPERATURE USING THE MIDDLE RANGE FUNCTION	FLOD2130
300	GO TO(31,32,33,34,35,36,37,38,39,40,41,42,43,44,45),N	FLOD2140
31	TI(1) = TIN1B(X)	FLOD2150
	GO TO 500	FLOD2160
32	TI(2) = TIN2B(X)	FLOD2170
	GO TO 500	FLOD2180
33	TI(3) = TIN3B(X)	FLOD2190
	GO TO 500	FLOD2200
34	TI(4) = TIN4B(X)	FLOD2210
	GO TO 500	FLOD2220
35	TI(5) = TIN5B(X)	FLOD2230
	GO TO 500	FLOD2240
36	TI(6) = TIN6B(X)	FLOD2250
	GO TO 500	FLOD2260
37	TI(7) = TIN7B(X)	FLOD2270
	GO TO 500	FLOD2280
38	TI(8) = TIN8B(X)	FLOD2290

	GO TO 500	FLOD2300
39	TI(9) = TIN9B(X)	FLOD2310
	GO TO 500	FLOD2320
40	TI(10) = TIN10B(X)	FLOD2330
	GO TO 500	FLOD2340
411	TI(11) = TIN11B(X)	FLOD2350
	GO TO 500	FLOD2360
412	TI(12) = TIN12B(X)	FLOD2370
	GO TO 500	FLOD2380
413	TI(13) = TIN13B(X)	FLOD2390
	GO TO 500	FLOD2400
414	TI(14) = TIN14B(X)	FLOD2410
	GO TO 500	FLOD2420
415	TI(15) = TIN15B(X)	FLOD2430
	GO TO 500	FLOD2440
CE	*****	FLOD2450
C/	EVALUATE THE INLET TEMPERATURE USING THE HIGH RANGE FUNCTION	FLOD2460
400	GO TO(41,42,43,44,45,46,47,48,49,50,511,512,513,514,515),N	FLOD2470
41	TI(1) = TIN1C(X)	FLOD2480
	GO TO 500	FLOD2490
42	TI(2) = TIN2C(X)	FLOD2500
	GO TO 500	FLOD2510
43	TI(3) = TIN3C(X)	FLOD2520
	GO TO 500	FLOD2530
44	TI(4) = TIN4C(X)	FLOD2540
	GO TO 500	FLOD2550
45	TI(5) = TIN5C(X)	FLOD2560
	GO TO 500	FLOD2570
46	TI(6) = TIN6C(X)	FLOD2580
	GO TO 500	FLOD2590
47	TI(7) = TIN7C(X)	FLOD2600
	GO TO 500	FLOD2610
48	TI(8) = TIN8C(X)	FLOD2620
	GO TO 500	FLOD2630
49	TI(9) = TIN9C(X)	FLOD2640
	GO TO 500	FLOD2650
50	TI(10) = TIN10C(X)	FLOD2660
	GO TO 500	FLOD2670
511	TI(11) = TIN11C(X)	FLOD2680
	GO TO 500	FLOD2690
512	TI(12) = TIN12C(X)	FLOD2700
	GO TO 500	FLOD2710
513	TI(13) = TIN13C(X)	FLOD2720
	GO TO 500	FLOD2730
514	TI(14) = TIN14C(X)	FLOD2740
	GO TO 500	FLOD2750

```

515 TI(15)=TIN15C(X) FLOD2760
CE ***** FLOD2770
C ===== FLOD2780
C CALCULATE HEAT TRANSFER COEFFICIENTS AND SPECIFIC HEATS FOR FLOD2790
C THE COOLANTS. FLOD2800
C ===== FLOD2810
C SET ALL COOLANT CONDUCTIVITIES TO ZERO FLOD2820
500 DO 510 K = KLS,KHS FLOD2830
      DO 510 J = JLS,JHS FLOD2840
      DO 510 I = ILS,IHS FLOD2850
      CONK(I,J,K)=1.E8 FLOD2860
      CONT(I,J,K)=1.E8 FLOD2870
      CONL(I,J,K)=1.E8 FLOD2880
510 CONTINUE FLOD2890
      N=IAMS(IPATH(N)) FLOD2900
C DETERMINE THE DIRECTION OF THE FLOW -'RADIAL,AXIAL,THETA'- FLOD2910
      GO TO(1000,2000,3000),M FLOD2920
C ***** FLOD2930
CB RADIAL COOLANT FLOW FLOD2940
C ***** FLOD2950
C DETERMINE THE BLOCK TYPE IN THE THETA DIRECTION. FLOD2960
1000 CALL BLKTYP(KHS,KLS,KGHS,KGLS,NTG,TYPET) FLOD2970
C DETERMINE THE BLOCK TYPE IN THE AXIAL DIRECTION. FLOD2980
      CALL BLATYP(JHS,JLS,JGHS,JGLS,NZG,TYPEZ) FLOD2990
C CALCULATE THE COOLANT PROPERTIES AT ALL LEVELS ALONG THE COOLANT FLOD3000
      DO 1500 I=ILS,IHS FLOD3010
C SET THE LEVEL AND LOCAL COOLANT TEMPERATURE FLOD3020
      PTRERP(I) FLOD3030
      DR=1(I,JLS,KLS) FLOD3040
C CALCULATE THE SPECIFIC HEAT AT THIS LEVEL FLOD3050
      KGPC(N,I)=SPEC(X) FLOD3060
C EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS FLOD3070
C WHICH EXTEND OVER THE ENTIRE THETA DISTANCE OF THE PROBLEM. FLOD3080
      IF(KGHS.EQ.NTG.AND.KGLS.EQ.1) GO TO 1210 FLOD3090
C CALCULATE THE HEAT TRANSFER COEFFICIENTS TO THE THETA WALLS FLOD3100
      DO 1200 J=JLS,JHS FLOD3110
      GO TO(1214,1212,1214,1216),TYPET FLOD3120
C CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH FLOD3130
C THETA BOUNDARIES OF THE COOLANT BLOCK. FLOD3140
1212 STA(1)=1/BBTH(KGLS,I,J) FLOD3150
      STA(2)=1/BBTL(KGHS,I,J) FLOD3160
      CONT(I,J,KLS)=1./HC((STA(1)+STA(2))*0.5) FLOD3170
      GO TO 1200 FLOD3180
C CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE FLOD3190
C THETA BOUNDARIES OF THE COOLANT BLOCK. FLOD3200
1214 STA(2)=1/BBTL(KGHS,I,J) FLOD3210

```

```

CONT(I,J,KHS)=1./HC(STA(2)) FLOD3220
IF(TYPET.NE.1) GO TO 1200 FLOD3230
1216 STA(1)=ZBOTH(KGLS,I,J) FLOD3240
CONT(I,J,KLS)=1./HC(STA(1)) FLOD3250
C HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED? FLOD3260
C .-'YES,NO'-' FLOD3270
1200 CONTINUE FLOD3280
C EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS FLOD3290
C WHICH EXTEND OVER THE ENTIRE AXIAL DISTANCE OF THE PROBLEM. FLOD3300
1210 IF(JGHS.EQ.N/6.AND.JGLS.EQ.1) GO TO 1500 FLOD3310
C CALCULATE THE HEAT TRANSFER COEFFICIENTS TO THE AXIAL WALLS FLOD3320
DO 1301 K=KLS,KHS FLOD3330
GO TO(1224,1222,1224,1226),TYPEZ FLOD3340
C CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH FLOD3350
C AXIAL BOUNDARIES OF THE COOLANT BLOCK. FLOD3360
1222 STA(1)=ZBOTH(JGLS,I,K) FLOD3370
STA(2)=ZBRTL(JGHS,I,K) FLOD3380
CONZ(I,JLS,K)=1./HC((STA(1)+STA(2))*5) FLOD3390
GO TO 1301 FLOD3400
C CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE FLOD3410
C AXIAL BOUNDARIES OF THE COOLANT BLOCK. FLOD3420
1224 STA(2)=ZBRTL(JGHS,I,K) FLOD3430
CONZ(I,JHS,K)=1./HC(STA(2)) FLOD3440
IF(TYPEZ.NE.1) GO TO 1301 FLOD3450
1226 STA(1)=ZBOTH(JGLS,I,K) FLOD3460
CONZ(I,JLS,K)=1./HC(STA(1)) FLOD3470
C HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED? FLOD3480
C .-'YES,NO'-' FLOD3490
1301 CONTINUE FLOD3500
C HAVE THE COOLANT PROPERTIES BEEN CALCULATED AT ALL LEVELS? FLOD3510
C .-'YES,NO'-' FLOD3520
1500 CONTINUE FLOD3530
CE ***** FLOD3540
RETURN FLOD3550
C ***** FLOD3560
CB AXIAL COOLANT FLOW FLOD3570
C ***** FLOD3580
C DETERMINE THE BLOCK TYPE IN THE THETA DIRECTION. FLOD3590
2000 CALL BLKTYP(KHS,KLS,KGHS,KGLS,NTG,TYPET) FLOD3600
C DETERMINE THE BLOCK TYPE IN THE RADIAL DIRECTION. FLOD3610
CALL BLKTYP(IHS,ILS,IGHS,IGLS,NRG,TYPEP) FLOD3620
C CALCULATE THE COOLANT PROPERTIES AT ALL LEVELS ALONG THE COOLANT FLOD3630
DO 2500 J=JLS,JHS FLOD3640
C SET THE LEVEL AND LOCAL COOLANT TEMPERATURE FLOD3650
FTZ=ZP(J) FLOD3660
URET(ILS,J,KLS) FLOD3670

```

C	CALCULATE THE SPECIFIC HEAT AT THIS LEVEL	FL0D3680
	RCPC(N,J)=SPEC(X)	FL0D3690
C	EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS	FL0D3700
C	WHICH EXTEND OVER THE ENTIRE THETA DISTANCE OF THE PROBLEM.	FL0D3710
	IF(KCHS.EQ.NTG.AND.KGLS.EQ.1) GO TO 2210	FL0D3720
C	CALCULATE THE HEAT TRANSFER COEFFICIENTS TO THE THETA WALLS	FL0D3730
	DO 2200 I=ILS,IHS	FL0D3740
	GO TO(2214,2212,2214,2216),TYPE1	FL0D3750
C	CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH	FL0D3760
C	THETA BOUNDARIES OF THE COOLANT BLOCK.	FL0D3770
2212	STA(1)=TBBTH(KGLS,I,J)	FL0D3780
	STA(2)=TBBTL(KGHS,I,J)	FL0D3790
	CON1(I,J,KLS)=1./HC((STA(1)+STA(2))*0.5)	FL0D3800
	GO TO 2200	FL0D3810
C	CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE	FL0D3820
C	THETA BOUNDARIES OF THE COOLANT BLOCK.	FL0D3830
2214	STA(2)=TBBTL(KGHS,I,J)	FL0D3840
	CON1(I,J,KHS)=1./HC(STA(2))	FL0D3850
	IF(TYPE1.NE.1) GO TO 2200	FL0D3860
2216	STA(1)=TBBTH(KGLS,I,J)	FL0D3870
	CON1(I,J,KLS)=1./HC(STA(1))	FL0D3880
C	HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED?	FL0D3890
C	.-'YES,NO'-	FL0D3900
2200	CONTINUE	FL0D3910
C	EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS	FL0D3920
C	WHICH EXTEND OVER THE ENTIRE RADIAL DISTANCE OF THE PROBLEM.	FL0D3930
2210	IF(IGH.S.EQ.NRG.AND.IGLS.EQ.1) GO TO 2500	FL0D3940
C	CALCULATE THE HEAT TRANSFER COEFFICIENTS TO THE RADIAL WALLS	FL0D3950
	DO 2300 K=KLS,KHS	FL0D3960
	GO TO(2224,2222,2224,2226),TYPE1	FL0D3970
C	CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH	FL0D3980
C	RADIAL BOUNDARIES OF THE COOLANT BLOCK.	FL0D3990
2222	STA(1)=RBBTH(IGLS,J,K)	FL0D4000
	STA(2)=RBBTL(IGHS,J,K)	FL0D4010
	CONK(ILS,J,K)=1./HC((STA(1)+STA(2))*0.5)	FL0D4020
	GO TO 2300	FL0D4030
C	CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE	FL0D4040
C	RADIAL BOUNDARIES OF THE COOLANT BLOCK.	FL0D4050
2224	STA(2)=RBBTL(IGHS,J,K)	FL0D4060
	CONK(IHS,J,K)=1./HC(STA(2))	FL0D4070
	IF(TYPE1.NE.1) GO TO 2300	FL0D4080
2226	STA(1)=RBBTH(IGLS,J,K)	FL0D4090
	CONK(ILS,J,K)=1./HC(STA(1))	FL0D4100
C	HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED?	FL0D4110
C	.-'YES,NO'-	FL0D4120
2300	CONTINUE	FL0D4130

```

C      HAVE THE COOLANT PROPERTIES BEEN CALCULATED AT ALL LEVELS?          FLOD4140
C      .='YES,NO'='                                                    FLOD4150
2500 CONTINUE                                                            FLOD4160
CE *****                                                                FLOD4170
RETURN                                                                    FLOD4180
C      *****                                                                FLOD4190
CB      THEIA COOLANT FLOW                                                FLOD4200
C      *****                                                                FLOD4210
C      DETERMINE THE BLOCK TYPE IN THE RADIAL DIRECTION.                FLOD4220
3000 CALL BLKTYP(IHS,ILS,IGHS,IGLS,NRG,TYPER)                            FLOD4230
C      DETERMINE THE BLOCK TYPE IN THE AXIAL DIRECTION.                FLOD4240
CALL BLKTYP(JHS,JLS,JGHS,JGLS,NZG,TYPEZ)                                FLOD4250
C      CALCULATE THE COOLANT PROPERTIES AT ALL LEVELS ALONG THE COOLANT FLOD4260
DO 3500 K=KLS,KHS                                                       FLOD4270
C      SET THE LEVEL AND LOCAL COOLANT TEMPERATURE                     FLOD4280
FIT=TP(K)                                                                FLOD4290
DK=1(ILS,JLS,K)                                                         FLOD4300
C      CALCULATE THE SPECIFIC HEAT AT THIS LEVEL                       FLOD4310
RCPG(N,K)=SPEC(X)                                                       FLOD4320
C      EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS FLOD4330
C      WHICH EXTEND OVER THE ENTIRE RADIAL DISTANCE OF THE PROBLEM.    FLOD4340
IF (IGHS.EQ.NRG.AND.IGLS.EQ.1) GO TO 3210                               FLOD4350
C      CALCULATE THE HEATTRANSFER COEFFICIENTS TO THE RADIAL WALLS    FLOD4360
DO 3200 J=JLS,JHS                                                       FLOD4370
GO TO (3214,3212,3214,3216),TYPER                                       FLOD4380
C      CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH        FLOD4390
C      RADIAL BOUNDARIES OF THE COOLANT BLOCK.                         FLOD4400
3212 STA(1)=RBBTH(IGLS,J,K)                                             FLOD4410
STA(2)=RBBTL(IGHS,J,K)                                                 FLOD4420
CONK(ILS,J,K)=1./HC((STA(1)+STA(2))*0.5)                                FLOD4430
GO TO 3200                                                                FLOD4440
C      CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE          FLOD4450
C      RADIAL BOUNDARIES OF THE COOLANT BLOCK.                         FLOD4460
3214 STA(2)=RBBTL(IGHS,J,K)                                             FLOD4470
CONK(IHS,J,K)=1./HC(STA(2))                                             FLOD4480
IF (TYPER.NE.1) GO TO 3200                                               FLOD4490
3216 STA(1)=RBBTH(IGLS,J,K)                                             FLOD4500
CONK(ILS,J,K)=1./HC(STA(1))                                             FLOD4510
C      HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED? FLOD4520
C      .='YES,NO'='                                                    FLOD4530
3200 CONTINUE                                                            FLOD4540
C      EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS FLOD4550
C      WHICH EXTEND OVER THE ENTIRE AXIAL DISTANCE OF THE PROBLEM.    FLOD4560
3210 IF (JGHS.EQ.NZG.AND.JGLS.EQ.1) GO TO 3500                          FLOD4570
C      CALCULATE THE HEATTRANSFER COEFFICIENTS TO THE AXIAL WALLS    FLOD4580
DO 3300 I = ILS,IHS                                                     FLOD4590

```



```

          GO TO(3224,3222,3224,3226),TYPEZ
C        CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH
C        AXIAL BOUNDARIES OF THE COOLANT BLOCK.
3222 STA(1)=ZBBTH(JGLS,I,K)
      STA(2)=ZBBTL(JGHS,I,K)
      CONZ(I,JLS,K)=1./HC((STA(1)+STA(2))*5)
      GO TO 3300
C        CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE
C        AXIAL BOUNDARIES OF THE COOLANT BLOCK.
3224 STA(2)=ZBBTL(JGHS,I,K)
      CONZ(I,JHS,K)=1./HC(STA(2))
      IF(TYPEZ.NE.1) GO TO 3300
3226 STA(1)=ZBBTH(JGLS,I,K)
      CONZ(I,JLS,K)=1./HC(STA(1))
C        HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED?
C        .='YES,NO'='
3300 CONTINUE
C        HAVE THE COOLANT PROPERTIES BEEN CALCULATED AT ALL LEVELS?
C        .='YES,NO'='
3500 CONTINUE
CE *****
      RETURN
CE *****
CE *****
CE *****
      FUNCTION SPEC(X)
C/      CALCULATE THE COOLANT SPECIFIC HEAT
C=====
1050 GO TO(1051,1052,1053,1054,1055,1056,1057,1058,1059,1060,1061,
      .1062,1063,1064,1065),N
1051 B=SPH1(X)
      GO TO 1070
1052 B=SPH2(X)
      GO TO 1070
1053 B=SPH3(X)
      GO TO 1070
1054 B=SPH4(X)
      GO TO 1070
1055 B=SPH5(X)
      GO TO 1070
1056 B=SPH6(X)
      GO TO 1070
1057 B=SPH7(X)
      GO TO 1070
1058 B=SPH8(X)
      GO TO 1070

```

```

FLOD4600
FLOD4610
FLOD4620
FLOD4630
FLOD4640
FLOD4650
FLOD4660
FLOD4670
FLOD4680
FLOD4690
FLOD4700
FLOD4710
FLOD4720
FLOD4730
FLOD4740
FLOD4750
FLOD4760
FLOD4770
FLOD4780
FLOD4790
FLOD4800
FLOD4810
FLOD4820
FLOD4830
FLOD4840
FLOD4850
FLOD4860
FLOD4870
FLOD4880
FLOD4890
FLOD4900
FLOD4910
FLOD4920
FLOD4930
FLOD4940
FLOD4950
FLOD4960
FLOD4970
FLOD4980
FLOD4990
FLOD5000
FLOD5010
FLOD5020
FLOD5030
FLOD5040
FLOD5050

```

```

1059 B=SPH9(X) FLOD5060
      GO TO 1070 FLOD5070
1060 B=SPH10(X) FLOD5080
      GO TO 1070 FLOD5090
1061 B=SPH11(X) FLOD5100
      GO TO 1070 FLOD5110
1062 B=SPH12(X) FLOD5120
      GO TO 1070 FLOD5130
1063 B=SPH13(X) FLOD5140
      GO TO 1070 FLOD5150
1064 B=SPH14(X) FLOD5160
      GO TO 1070 FLOD5170
1065 B=SPH15(X) FLOD5180
1070 SPEC=B FLOD5190
      RETURN FLOD5200
CE ***** FLOD5210
CE ***** FLOD5220
CE ***** FLOD5230
      FUNCTION HC(X) FLOD5240
C/ CALCULATE THE HEAT TRANSFER COEFFICIENT FLOD5250
C===== FLOD5260
      ST=X FLOD5270
1300 GO TO (1310,1320,1330,1340,1350,1360,1370,1380,1390,1400,1410,
      .1420,1430,1440,1450),N FLOD5280
C===== FLOD5290
1310 RE=REYN1(X) FLOD5310
      IA=SELECT(N) FLOD5320
      GO TO (1311,1312,1313),IA FLOD5330
1311 B=H1A(X) FLOD5340
      GO TO 1399 FLOD5350
1312 B=H1B(X) FLOD5360
      GO TO 1399 FLOD5370
1313 B=H1C(X) FLOD5380
      GO TO 1399 FLOD5390
C===== FLOD5400
1320 RE=REYN2(X) FLOD5410
      IA=SELECT(N) FLOD5420
      GO TO (1321,1322,1323),IA FLOD5430
1321 B=H2A(X) FLOD5440
      GO TO 1399 FLOD5450
1322 B=H2B(X) FLOD5460
      GO TO 1399 FLOD5470
1323 B=H2C(X) FLOD5480
      GO TO 1399 FLOD5490
C===== FLOD5500
1330 RE=REYN3(X) FLOD5510

```

	IA=SELECT(N)	FL0D520
	GO TO (1331,1332,1333),IA	FL0D530
1331	B=H3A(X)	FL0D540
	GO TO 1399	FL0D550
1332	B=H3B(X)	FL0D560
	GO TO 1399	FL0D570
1333	B=H3C(X)	FL0D580
	GO TO 1399	FL0D590
C=====		FL0D5600
1340	RE=KEYH4(X)	FL0D5610
	IA=SELECT(N)	FL0D5620
	GO TO (1341,1342,1343),IA	FL0D5630
1341	B=H4A(X)	FL0D5640
	GO TO 1399	FL0D5650
1342	B=H4B(X)	FL0D5660
	GO TO 1399	FL0D5670
1343	B=H4C(X)	FL0D5680
	GO TO 1399	FL0D5690
C=====		FL0D5700
1350	RE=KEYH5(X)	FL0D5710
	IA=SELECT(N)	FL0D5720
	GO TO (1351,1352,1353),IA	FL0D5730
1351	B=H5A(X)	FL0D5740
	GO TO 1399	FL0D5750
1352	B=H5B(X)	FL0D5760
	GO TO 1399	FL0D5770
1353	B=H5C(X)	FL0D5780
	GO TO 1399	FL0D5790
C=====		FL0D5800
1360	RE=KEYH6(X)	FL0D5810
	IA=SELECT(N)	FL0D5820
	GO TO (1361,1362,1363),IA	FL0D5830
1361	B=H6A(X)	FL0D5840
	GO TO 1399	FL0D5850
1362	B=H6B(X)	FL0D5860
	GO TO 1399	FL0D5870
1363	B=H6C(X)	FL0D5880
	GO TO 1399	FL0D5890
C=====		FL0D5900
1370	RE=KEYH7(X)	FL0D5910
	IA=SELECT(N)	FL0D5920
	GO TO (1371,1372,1373),IA	FL0D5930
1371	B=H7A(X)	FL0D5940
	GO TO 1399	FL0D5950
1372	B=H7B(X)	FL0D5960
	GO TO 1399	FL0D5970

1373	B=H7C(X)	FLOD5980
	GO TO 1399	FLOD5990
C=====		
1380	RE=KEYN8(X)	FLOD6010
	IA=SELECT(N)	FLOD6020
	GO TO (1381,1382,1383),IA	FLOD6030
1381	B=H8A(X)	FLOD6040
	GO TO 1399	FLOD6050
1382	B=H8B(X)	FLOD6060
	GO TO 1399	FLOD6070
1383	B=H8C(X)	FLOD6080
	GO TO 1399	FLOD6090
C=====		
1390	RE=KEYN9(X)	FLOD6110
	IA=SELECT(N)	FLOD6120
	GO TO (1391,1392,1393),IA	FLOD6130
1391	B=H9A(X)	FLOD6140
	GO TO 1399	FLOD6150
1392	B=H9B(X)	FLOD6160
	GO TO 1399	FLOD6170
1393	B=H9C(X)	FLOD6180
	GO TO 1399	FLOD6190
C=====		
1400	RE=KEYN10(X)	FLOD6210
	IA=SELECT(N)	FLOD6220
	GO TO (1401,1402,1403),IA	FLOD6230
1401	B=H10A(X)	FLOD6240
	GO TO 1399	FLOD6250
1402	B=H10B(X)	FLOD6260
	GO TO 1399	FLOD6270
1403	B=H10C(X)	FLOD6280
	GO TO 1399	FLOD6290
C=====		
1410	RE=KEYN11(X)	FLOD6310
	IA=SELECT(N)	FLOD6320
	GO TO (1411,1412,1413),IA	FLOD6330
1411	B=H11A(X)	FLOD6340
	GO TO 1399	FLOD6350
1412	B=H11B(X)	FLOD6360
	GO TO 1399	FLOD6370
1413	B=H11C(X)	FLOD6380
	GO TO 1399	FLOD6390
C=====		
1420	RE=KEYN12(X)	FLOD6410
	IA=SELECT(N)	FLOD6420
	GO TO (1421,1422,1423),IA	FLOD6430

```

1421 B=H12A(X)                                FLOD6440
      GO TO 1399                                FLOD6450
1422 B=H12B(X)                                FLOD6460
      GO TO 1399                                FLOD6470
1423 B=H12C(X)                                FLOD6480
      GO TO 1399                                FLOD6490
C=====
1430 RE=REYN13(X)                              FLOD6510
      IA=SELECT(N)                              FLOD6520
      GO TO(1431,1432,1433),IA                 FLOD6530
1431 B=H13A(X)                                FLOD6540
      GO TO 1399                                FLOD6550
1432 B=H13B(X)                                FLOD6560
      GO TO 1399                                FLOD6570
1433 B=H13C(X)                                FLOD6580
      GO TO 1399                                FLOD6590
C=====
1440 RE=REYN14(X)                              FLOD6610
      IA=SELECT(N)                              FLOD6620
      GO TO(1441,1442,1443),IA                 FLOD6630
1441 B=H14A(X)                                FLOD6640
      GO TO 1399                                FLOD6650
1442 B=H14B(X)                                FLOD6660
      GO TO 1399                                FLOD6670
1443 B=H14C(X)                                FLOD6680
      GO TO 1399                                FLOD6690
C=====
1450 RE=REYN15(X)                              FLOD6710
      IA=SELECT(N)                              FLOD6720
      GO TO(1451,1452,1453),IA                 FLOD6730
1451 B=H15A(X)                                FLOD6740
      GO TO 1399                                FLOD6750
1452 B=H15B(X)                                FLOD6760
      GO TO 1399                                FLOD6770
1453 B=H15C(X)                                FLOD6780
C=====
1399 C=B                                        FLOD6800
      IF (C.EQ.0.0) C=1.0E-6                   FLOD6810
      HC=C                                       FLOD6820
      RETURN                                     FLOD6830
CE *****FLOD6840
CE *****FLOD6850
CE *****FLOD6860
      INTLGER FUNCTION SELECT(N)                FLOD6870
C=====
C DOES THE REYNOLDS NUMBER LIE WITHIN THE SPECIFIED RANGE? FLOD6880
C

```

```

C      . -'NO,YES'-                                FLOD6900
      IF(RE.LT,RLIM1(N).OR,RE.GT,RLIM4(N))GO TO 500 FLOD6910
C      DOES THE REYNOLDS NUMBER LIE IN THE LOW RANGE? -'YES,NO'- FLOD6920
      IF(RE.LT,RLIM2(N))GO TO 100                   FLOD6930
C      DOES THE REYNOLDS NUMBER LIE IN THE MIDDLE RANGE? -'YES,NO'- FLOD6940
      IF(RE.LT,RLIM3(N))GO TO 200                   FLOD6950
C      UPPER RANGE HEAT TRANSFER CORRELATION        FLOD6960
      SELECT=3                                       FLOD6970
      RETURN                                         FLOD6980
C      LOWER RANGE HEAT TRANSFER ROUTINE            FLOD6990
100  SELECT=1                                       FLOD7000
      RETURN                                         FLOD7010
C      MIDDLE RANGE HEAT TRANSFER CORRELATION      FLOD7020
200  SELECT=2                                       FLOD7030
      RETURN                                         FLOD7040
C=====FLOD7050
C      THE REYNOLDS NUMBER LIES OUTSIDE THE SPECIFIED RANGE FLOD7060
500  IERROR(1)=IFLODA(1)                             FLOD7070
      IERROR(2)=IFLODA(4)                             FLOD7080
      CALL ERROR2                                     FLOD7090
C=====FLOD7100
      END                                           FLOD7110

```

```

SUBROUTINE BLKTYP(XHS,XLS,XGHS,XGLS,NXG,TYPE)                                BLKT 10
=====BLKT 20
C DETERMINE COOLANT BLOCK TYPE WITH RESPECT TO BOUNDARIES ALONG           BLKT 30
C WHICH HEAT TRANSFER COEFFICIENTS ARE TO BE ASSIGNED.                   BLKT 40
C   TYPE=1  ASSIGN SEPARATE VALUES ALONG BOTH BOUNDARIES.               BLKT 50
C   TYPE=2  ASSIGN AN AVERAGE OF THE SEPARATE VALUES ALONG             BLKT 60
C           BOTH BOUNDARIES.                                             BLKT 70
C   TYPE=3  ASSIGN A VALUE ALONG THE HIGH BOUNDARY ONLY.                BLKT 80
C   TYPE=4  ASSIGN A VALUE ALONG THE LOW BOUNDARY ONLY.                  BLKT 90
C =====BLKT 100
C   INTEGER      XHS, XLS, XGHS, XGLS, TYPE                               PLKT 110
C =====BLKT 120
C =====BLKT 130
C   TYPE=0                                             BLKT 140
C IS THIS AN INTERNAL COOLANT BLOCK ?                               BLKT 150
C IF(XGHS.NE.XGLS) GO TO 100                                       BLKT 160
C SHOULD THE HEAT TRANSFER COEFFICIENT BE CALCULATED FOR THE HIGH     BLKT 170
C OR THE LOW BOUNDARY OF THE COOLANT BLOCK ?                       PLKT 180
C IF(XGHS.EQ.1) GO TO 110                                           PLKT 190
C GO TO 120                                                         BLKT 200
C IS THE INTERNAL COOLANT BLOCK ADJACENT TO AN EXTERNAL COOLANT ON    BLKT 210
C EITHER OF THE TWO BOUNDARIES BEING CONSIDERED ?                 PLKT 220
100 IF(XGHS.EQ.NXG) GO TO 120                                       BLKT 230
C   IF(XGLS.EQ.1) GO TO 110                                         PLKT 240
C IS THE COOLANT BLOCK ONLY ONE POINT WIDE IN THE DIRECTION         BLKT 250
C BEING CONSIDERED ?                                             BLKT 260
C IF(XLS.NE.XHS) GO TO 110                                         BLKT 270
C   TYPE=2                                                         BLKT 280
C   GO TO 130                                                       BLKT 290
110 TYPE=3                                                         BLKT 300
C   IF(XGLS.EQ.1) GO TO 130                                         BLKT 310
C   TYPE=1                                                         PLKT 320
C   GO TO 130                                                       BLKT 330
120 TYPE=4                                                         BLKT 340
130 RETURN                                                         BLKT 350
C   END                                                             BLKT 360

```

```

FUNCTION COOL1 (X)
ENTRY SPH1(X)
COOL1 = 1.0
GO TO 10
ENTRY REYN1(X)
COOL1 = 0.0
GO TO 10
ENTRY H1A(X)
COOL1 = 0.0
GO TO 10
ENTRY H1B(X)
COOL1 = 0.0
GO TO 10
ENTRY H1C(X)
COOL1 = 0.0
GO TO 10
ENTRY FLO1A(X)
COOL1 = 1000000.0
GO TO 10
ENTRY FLO1B(X)
COOL1 = 0.0
GO TO 10
ENTRY FLO1C(X)
COOL1 = 0.0
GO TO 10
ENTRY TIN1A(X)
COOL1 = 460.0
GO TO 10
ENTRY TIN1B(X)
COOL1 = 0.0
GO TO 10
ENTRY TIN1C(X)
COOL1 = 0.0
10 RETURN
END

```

```

COOL  0
COOL  20
COOL  30
COOL  40
COOL  50
COOL  60
COOL  70
COOL  80
COOL  90
COOL 100
COOL 110
COOL 120
COOL 130
COOL 140
COOL 150
COOL 160
COOL 170
COOL 180
COOL 190
COOL 200
COOL 210
COOL 220
COOL 230
COOL 240
COOL 250
COOL 260
COOL 270
COOL 280
COOL 290
COOL 300
COOL 310
COOL 320
COOL 330
COOL 340
COOL 350

```


FUNCTION COOL2 (X)	COOL 10
ENTRY SPH2(X)	COOL 20
COOL2 = 1.0	COOL 30
GO TO 10	COOL 40
ENTRY REYN2(X)	COOL 50
COOL2 = 0.0	COOL 60
GO TO 10	COOL 70
ENTRY H2A(X)	COOL 80
COOL2 = 0.0	COOL 90
GO TO 10	COOL 100
ENTRY H2B(X)	COOL 110
COOL2 = 0.0	COOL 120
GO TO 10	COOL 130
ENTRY H2C(X)	COOL 140
COOL2 = 0.0	COOL 150
GO TO 10	COOL 160
ENTRY FLO2A(X)	COOL 170
COOL2 = 1000000.0	COOL 180
GO TO 10	COOL 190
ENTRY FLO2B(X)	COOL 200
COOL2 = 0.0	COOL 210
GO TO 10	COOL 220
ENTRY FLO2C(X)	COOL 230
COOL2 = 0.0	COOL 240
GO TO 10	COOL 250
ENTRY TIN2A(X)	COOL 260
COOL2 = 460.0	COOL 270
GO TO 10	COOL 280
ENTRY TIN2B(X)	COOL 290
COOL2 = 0.0	COOL 300
GO TO 10	COOL 310
ENTRY TIN2C(X)	COOL 320
COOL2 = 0.0	COOL 330
10 RETURN	COOL 340
END	COOL 350

NOTE:

The functions COOL3 through COOL15 are omitted since they are all of the form illustrated in the preceding listings of COOL1 and COOL2.

```

SUBROUTINE CONDOC                                COND 10
INCLUDE      COMDIM                               COND 20
C =====COND 30
CB CALCULATE THE AVERAGE CONDUCTIVITIES BETWEEN POINTS. RADIATION COND 40
C ACROSS GAPS AND INTERNAL COOLANTS IS INCLUDED. COND 50
C =====COND 60
C ERROR MESSAGES COND 70
C   CONDOC1 , CONDOC2 , CONDOC3 COND 80
C   THESE MESSAGES INDICATE THAT RADIATION BETWEEN THE COND 90
C   BOUNDARIES OF AN INTERNAL COOLANT BLOCK WAS NEGLECTED COND 100
C   IN ORDER TO COMPLETE AN ITERATION. A SMALLER TIME COND 110
C   STEP SHOULD BE USED IF THE NEGLECT OF RADIATION IS COND 120
C   SIGNIFICANT. COND 130
C   COND 140
C   CONDOC1 RADIATION WAS NEGLECTED BETWEEN RADIAL GAPLINES IGL COND 150
C   AND IGH AT AXIAL POINT LEVEL J, THETA POINT LEVEL K. COND 160
C   CONDOC2 RADIATION WAS NEGLECTED BETWEEN AXIAL GAPLINES JGL COND 170
C   AND JKH AT RADIAL POINT LEVEL I, THETA POINT LEVEL K. COND 180
C   CONDOC3 RADIATION WAS NEGLECTED BETWEEN THETA GAPLINES KGL COND 190
C   AND KGH AT RADIAL POINT LEVEL I, AXIAL POINT LEVEL J. COND 200
C =====COND 210
C LOGICAL      NOMES COND 220
C DIMENSION    ICOND(13) COND 230
C DATA(ICOND(I), I=1,13) / 1H1,5H IGL=,5H IGH=,3H J=,3H K=,1H2, COND 240
C 5H JGL=,5H JKH=,3H I=,1H3,5H KGL=,5H KGH=,3H K= / COND 250
100 FORMAT(16H ERROR AT CONDOC, A1, 17H ON ITERATION NO., I4, 3H AT, A5, I2, COND 260
C 5H, I2, A3, I2, A3, I2) - COND 270
C =====COND 280
C SET THE CORRECT ENTRY FOR MADATA COND 290
C GAS = .TRUE. COND 300
C =====COND 310
C TDMIN = 0.10 COND 320
C NITER = NITER + 1 COND 330
C NOMES = .FALSE. COND 340
C IF(.NOT.SW(10)) GO TO 105 COND 350
C IF(CS1.LT.1.0) NOMES = .TRUE. COND 360
105 IF(FIRST) NOMES = .TRUE. COND 370
C =====COND 380
C CALCULATE THE AVERAGE RADIAL CONDUCTIVITIES BETWEEN POINTS. COND 390
C =====COND 400
C CALCULATE THE RADIAL CONDUCTIVITIES IN EACH RADIAL PLANE. COND 410
C DO 340 I=1,IM COND 420
C 16 = IGR(I) COND 430
C =====COND 440
C CALCULATE THE RADIAL CONDUCTIVITIES IN EACH AXIAL COLUMN IN THE COND 450

```

```

C      PLANE. COND 460
C      DO 330 J=2,JM COND 470
C      =====COND 480
C      CALCULATE THE RADIAL CONDUCTIVITIES IN THE COLUMN. COND 490
C      DO 320 K=2,KM COND 500
C      DOES THE GRIDLINE HAVE ANYWHERE A GAP OR A COOLANT ADJACENT TO IT COND 510
C      ? -'NO,YES'- COND 520
C      IF (IG.EQ.0) GO TO 110 COND 530
C      =====COND 540
C      IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS COND 550
C      POINT ? -'NO,YES'- COND 560
C      IF (GAPR(IG,J,K).EQ.0.0) GO TO 110 COND 570
C      IS IT A GAP OR A COOLANT WHICH IS ADJACENT ? -'COOLANT,GAP'- COND 580
C      IF (GAPR(IG,J,K).LT.0.0) GO TO 130 COND 590
C      =====COND 600
CB     CALCULATE THE AVERAGE CONDUCTIVITY WITH A GAP PRESENT COND 610
C      NI = MATRG(IG, J, K) COND 620
C      TH = RRBTH(IG, J, K) COND 630
C      TB = RRBTL(IG, J, K) COND 640
C      FTR = RP(I) COND 650
C      FTZ = ZP(J) COND 660
C      FTI=TP(K) COND 670
C      CALCULATE THE LOCAL GAS CONDUCTIVITY. COND 680
C      CALL MAUATA COND 690
C      BX = REMH(IG, J, K) + REML(IG, J, K) - 1.0 COND 700
C      CX = .1713E-4*(TH**2 + TB**2)*(TH + TB)/BX COND 710
C      X1 = CONR(I, J, K)*RATIOB(IG, J, K) COND 720
C      X2 = 1./((GK/GAPR(IG, J, K) + CX)*RATIOH(IG)) COND 730
C      X3 = CONR(I + 1, J, K) COND 740
C      PRESERVE THE DATA REQUIRED FOR CALCULATING RADIAL GAP BOUNDARY COND 750
C      TEMPERATURES BY STORING IT IN THE REM-RBBL LOCATIONS FOR THE COND 760
C      CURRENT GAPLINE. COND 770
C      REML(IG, J, K) = CONR(I, J, K) COND 780
C      REMH(IG, J, K) = CONR(I + 1, J, K) COND 790
C      RRBTL(IG, J, K) = BX COND 800
C      RRBTH(IG, J, K) = GK COND 810
C      KR(I, J, K) = 1./((X1 + X2 + X3)/RATIOK(IG)) COND 820
C      GO TO 310 COND 830
CE     *****COND 840
CE     *****COND 850
C      ARE THE CONDUCTIVITIES ON BOTH SIDES OF THE GRID PLANE IDENTICAL COND 860
C      ? -'NO,YES'- COND 870
C      110 IF (CONR(I,J,K).NE.CONR(I+1,J,K)) GO TO 120 COND 880
C      ASSIGN ONE OF THEM AS THE AVERAGE CONDUCTIVITY COND 890
C      KR(I, J, K) = 1.0/CONR(I, J, K) COND 900
C      GO TO 310 COND 910

```

```

C      CALCULATE THE AVERAGE CONDUCTIVITY WHEN NO GAP OR COOLANT IS      COND 920
C      PRESENT                                                                COND 930
120  KK(I, J, K) = 1.0/(CONR(I + 1, J, K) + (CONR(I, J, K) - CONR(I + 1,   COND 940
    , J, K))*RLN(I))                                                         COND 950
    GO TO 310                                                                COND 960
CB     CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ADJACENT        COND 970
C      IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GRIDPLANE? -NO,YES'-  COND 980
130  IF (GAPR(IG,J,K).LT.(-1.5E-10)) GO TO 210                             COND 990
C      CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE HIGH    COND 1000
C      INDEX SIDE.                                                         COND 1010
C      EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.                COND 1020
    IF (IG.EQ.NRG) GO TO 190                                               COND 1030
C      FIND THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.      COND 1040
    IG1 = IG + 1                                                           COND 1050
    DO 150 I1=IG1,NRG                                                      COND 1060
    IF (GAPR(I1,J,K).LT.(-1.5E-10)) GO TO 140                             COND 1070
    GO TO 150                                                                COND 1080
C      EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT THE  COND 1090
C      HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.              COND 1100
140  IF (GAPR(I1,J,K).LT.(-2.5E-10)) GO TO 190                           COND 1110
    IGH = I1                                                                COND 1120
    GO TO 160                                                                COND 1130
150  CONTINUE                                                             COND 1140
160  IF (ABS(REMH(IG,J,K)).GT..999E6.OR.ABS(REML(IGH,J,K)).GT..999E6) GO  COND 1150
    TO 170                                                                  COND 1160
    IF (ABS(T(I,J,K)-T(I+1,J,K)).LT.TDMIN) GO TO 180                      COND 1170
    RAD = 0.1713E-8*(RBBTH(IG, J, K)**4 - RBBTL(IGH, J, K)**4)/(REMH(   COND 1180
    , IG, J, K) + REML(IGH, J, K) - 1.0)                                  COND 1190
    HX = CONR(I + 1, J, K)*(T(I, J, K) - T(I + 1, J, K) - RAD*CONR(I,   COND 1200
    , J, K)*DELR(I)*RLN(I))/(T(I, J, K) - T(I + 1, J, K) + RAD*CONR(I  COND 1210
    , + 1, J, K))                                                         COND 1220
    MATKG(IG, J, K) = 200                                                  COND 1230
    MATKG(IGH, J, K) = 200                                                COND 1240
    GO TO 200                                                                COND 1250
170  MATKG(IG, J, K) = 100                                                COND 1260
    MATKG(IGH, J, K) = 100                                                COND 1270
    GO TO 190                                                                COND 1280
180  MATKG(IG, J, K) = 300                                                COND 1290
    MATKG(IGH, J, K) = 300                                                COND 1300
190  HX = CONR(I + 1, J, K)                                               COND 1310
200  KR(I, J, K) = 1.0/(CONR(I, J, K)*RLN(I) + HX*RATIOC(IG))          COND 1320
    GO TO 310                                                                COND 1330
C      ARE TWO COOLANTS ADJACENT? -NO,YES'-                                COND 1340
210  IF (GAPR(IG,J,K).GT.(-2.5E-10)) GO TO 220                          COND 1350
C      DO NOT ALLOW HEAT TRANSFER BETWEEN COOLANTS                       COND 1360
    KR(I, J, K) = 1.0E-10                                                 COND 1370

```

```

GO TO 310
C CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE COND1380
C LOW INDEX SIDE. COND1390
C EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS. COND1400
220 IF (IG.EQ.1) GO TO 290 COND1410
C FIND THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK. COND1420
IG1 = IG - 1 COND1430
DO 240 I1=1,IG1 COND1440
IG2 = IG - I1 COND1450
IF (GAPR(IG2,J,K).LT.(-.5E-10)) GO TO 230 COND1460
GO TO 240 COND1470
C EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT THE COND1480
C LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK. COND1490
230 IF (GAPR(IG2,J,K).LT.(-.2.5E-10)) GO TO 290 COND1500
IGL = IG2 COND1510
GO TO 250 COND1520
240 CONTINUE COND1530
250 IF (.NOT.(MATRG(IG,J,K).EQ.100.OR.MATRG(IG,J,K).EQ.300)) GO TO 290 COND1540
IF (ABS(T(I,J,K)-T(I+1,J,K)).LT.TDMIN) GO TO 260 COND1550
RAD = 0.1713E-8*(RBBTH(IGL,J,K)**4 - RBBTL(IG,J,K)**4)/(REMH( COND1560
* IGL,J,K) + REML(IG,J,K) - 1.0) COND1570
HX = CONR(I,J,K)*(T(I+1,J,K) - T(I,J,K) + CONR(I+1,J,K)COND1580
* )*RAD+RATIOC(IG,J,K)*DELR(I)/RATIOC(IG))/(T(I+1,J,K) - T(I COND1590
* ,J,K) - CONR(I,J,K)*RAD*RATIOC(IG,J,K)) COND1600
GO TO 300 COND1610
260 MATRG(IG,J,K) = 300 COND1620
MATRG(IGL,J,K) = 300 COND1630
DO 270 IX=1,IM COND1640
IF (IGR(IX).NE.IGL) GO TO 270 COND1650
IPL = 1X COND1660
GO TO 260 COND1670
270 CONTINUE COND1680
C RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE COND1690
C PRECEDING AND PRESERVED IN THE REM-RBBT ARRAYS. COND1700
C CONL IS CONR(IPL,J,K) COND1710
280 CONL=REML(IGL,J,K) COND1720
C CONH IS CONR(IPL+1,J,K) COND1730
CONH = RBBTL(IGL,J,K) COND1740
HX=CONH COND1750
KR(IPL,J,K) = 1.0/(CONL*RLN(IPL) + HX*RATIOC(IGL)) COND1760
290 HX = CONR(I,J,K) COND1770
300 KR(I,J,K) = 1.0/(CONR(I+1,J,K)/RATIOC(IG) + HX*RATIOC(IG)) COND1780
IF (MATRG(IG,J,K).NE.300) GO TO 310 COND1790
IF (NOMES) GO TO 310 COND1800
IERROR(1) = IERROR(1) + 1 COND1810
WRITE (6,100) ICOND(1),NNITER,ICOND(2),IGL,ICOND(3),IG,ICOND(4),J,ICOND(5)
COND1820
COND1830

```

```

1ICOND(13),K
CE *****COND1840
C 310 CONTINUE *****COND1850
C HAVE ALL POINTS IN THE COLUMN BEEN CONSIDERED ? -'YES,NO'- COND1860
C 320 CONTINUE *****COND1870
C HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'- COND1880
C 330 CONTINUE *****COND1890
C HAVE ALL PLANES BEEN CONSIDERED ? -'YES,NO'- COND1900
C 340 CONTINUE *****COND1910
CE *****COND1920
C *****COND1930
C *****COND1940
C *****COND1950
C CALCULATE THE AVERAGE AXIAL CONDUCTIVITIES BETWEEN POINTS. COND1960
C *****COND1970
C CALCULATE THE AXIAL CONDUCTIVITIES IN EACH AXIAL PLANE. COND1980
C DO 580 J=1,JM COND1990
  JG = JGZ(J) COND2000
C *****COND2010
C CALCULATE THE AXIAL CONDUCTIVITIES IN EACH RADIAL COLUMN IN THE COND2020
C PLANE. COND2030
C DO 570 I=2,IM COND2040
C *****COND2050
C CALCULATE THE AXIAL CONDUCTIVITIES IN THE COLUMN. COND2060
C DO 560 K=2,KM COND2070
  DOES THE GRIDLINE HAVE ANYWHERE A GAP OR A COOLANT ADJACENT TO IT COND2080
  ? -'NO,YES'- COND2090
  IF (JG.LT.0) GO TO 350 COND2100
  *****COND2110
  IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS COND2120
  POINT ? -'NO,YES'- COND2130
  IF (GAPZ(JG,I,K).EQ.0.0) GO TO 350 COND2140
  IS IT A GAP OR A COOLANT WHICH IS ADJACENT ? -'COOLANT,GAP'- COND2150
  IF (GAPZ(JG,I,K).LT.0.0) GO TO 370 COND2160
  *****COND2170
  CB CALCULATE THE AVERAGE CONDUCTIVITY WITH A GAP PRESENT COND2180
  NI = MATZG(JG, I, K) COND2190
  TH = ZBOTH(JG, I, K) COND2200
  TB = ZBOTL(JG, I, K) COND2210
  FIR = RP(I) COND2220
  FLZ = ZP(J) COND2230
  FIT = TP(K) COND2240
  C CALCULATE THE LOCAL GAS CONDUCTIVITY. COND2250
  CALL MAUATA COND2260
  BX = ZEMH(JG, I, K) + ZEML(JG, I, K) - 1.0 COND2270
  CX = .1713E-R*(TH**2 + TB**2)*(TH + TB)/BX COND2280
  XI = CONZ(I, J, K)*ZATIOB(JG, I, K) COND2290

```

```

X2=1./((GK/GAPZ(JG,1,K)+CX)*ZATIOH(JG))          COND2300
X3 = CONZ(I, J + 1, K)                            COND2310
C PRESERVE THE DATA REQUIRED FOR CALCULATING AXIAL GAP BOUNDARY COND2320
C TEMPERATURES BY STORING IT IN THE ZEM-7BBT LOCATIONS FOR THE COND2330
C CURRENT GAPLINE.                                COND2340
ZEMH(JG, I, K) = CONZ(I, J, K)                    COND2350
ZEMH(JG, I, K) = CONZ(I, J + 1, K)                COND2360
ZBB1L(JG, I, K) = BX                               COND2370
ZBB1H(JG, I, K) = GK                               COND2380
KZ(I, J, K) = 1./((X1 + X2 + X3)/ZATIOK(JG))      COND2390
GO TO 550                                          COND2400
CE *****COND2410
CE *****COND2420
C ARE THE CONDUCTIVITIES ON BOTH SIDES OF THE GRID PLANE IDENTICAL COND2430
C ? -'NO,YES'=-                                  COND2440
350 IF (CONZ(I,J,K).NE.CONZ(I,J+1,K)) GO TO 360    COND2450
C ASSIGN ONE OF THEM AS THE AVERAGE CONDUCTIVITY COND2460
KZ(I, J, K) = 1.0/CONZ(I, J, K)                  COND2470
GO TO 550                                          COND2480
C CALCULATE THE AVERAGE CONDUCTIVITY WHEN NO GAP OR COOLANT IS COND2490
C PRESENT                                         COND2500
360 KZ(I, J, K) = 1.0/(CONZ(I, J + 1, K) + (CONZ(I, J, K) - COND2510
CONZ(I, J + 1, K))*ZLN(J))                        COND2520
GO TO 550                                          COND2530
CB CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ADJACENT COND2540
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GRIDPLANE?'-NO,YES'=- COND2550
370 IF (GAPZ(JG,I,K).LT.(-1.5E-10)) GO TO 450    COND2560
C CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE HIGH COND2570
C INDEX SIDE.                                     COND2580
DXT = ZL(J) - ZH(J)                               COND2590
C EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS. COND2600
IF (JG.LQ.NZG) GO TO 430                          COND2610
C FIND THE HIGH AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK. COND2620
JG1 = JG + 1                                       COND2630
DO 390 JJ=JG1,NZG                                  COND2640
IF (GAPZ(JJ,I,K).LT.(-1.5E-10)) GO TO 380        COND2650
GO TO 390                                          COND2660
C EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT COND2670
C THE HIGH AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK. COND2680
380 IF (GAPZ(JJ,I,K).LT.(-2.5E-10)) GO TO 430    COND2690
JGH = JJ                                           COND2700
GO TO 400                                          COND2710
390 CONTINUE                                       COND2720
400 IF (ABS(ZEMH(JG,I,K)).GT..999E6.OR.ABS(ZEMH(JGH,I,K)).GT..999E6) GCOND2730
GO TO 410                                          COND2740
IF (ABS(T(I,J,K)-T(I,J+1,K)).LT.TDMIN) GO TO 420 COND2750

```

```

RAD = 0.1713E-8*(ZBBTH(JG, I, K)**4 - ZBBTL(JGH, I, K)**4)/(ZEMH( COND2760
' JG, I, K) + ZEML(JGH, I, K) - 1.0) COND2770
HX = CONZ(I, J + 1, K)*(T(I, J, K) - T(I, J + 1, K) - RAD*CONZ(I, COND2780
' J, K)*DXT)/(T(I, J, K) - T(I, J + 1, K) + RAD*CONZ(I, J + 1, K)) COND2790
MATZG(JG, I, K) = 200 COND2800
MATZG(JGH, I, K)=200 COND2810
GO TO 440 COND2820
410 MATZG(JG, I, K) = 100 COND2830
MATZG(JGH, I, K)=100 COND2840
GO TO 430 COND2850
420 MATZG(JG, I, K) = 300 COND2860
MATZG(JGH, I, K)=300 COND2870
430 HX = CONZ(I, J + 1, K) COND2880
440 KZ(I, J, K)=DELZ(J)/(CONZ(I, J, K)*DXT+HX) COND2890
GO TO 550 COND2900
C ARE TWO COOLANTS ADJACENT? -1=NO, YES=- COND2910
450 IF (GAPZ(JG, I, K).GT.(-2.5E-10)) GO TO 460 COND2920
C DO NOT ALLOW HEAT TRANSFER BETWEEN COOLANTS COND2930
KZ(I, J, K) = 1.0E-10 COND2940
GO TO 550 COND2950
C CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE LOW COND2960
C INDLX SIDE. COND2970
460 DXT = ZP(J + 1) - ZL(J) COND2980
C EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS. COND2990
IF (JG.EQ.1) GO TO 530 COND3000
C FIND THE LOW AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK. COND3010
JG1 = JG - 1 COND3020
DO 480 JJ=1, JG1 COND3030
JG2 = JG - JJ COND3040
IF (GAPZ(JG2, I, K).LT.(-.5E-10)) GO TO 470 COND3050
GO TO 480 COND3060
C EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT COND3070
C THE LOW AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK. COND3080
470 IF (GAPZ(JG2, I, K).LT.(-2.5E-10)) GO TO 530 COND3090
JG1 = JG2 COND3100
GO TO 490 COND3110
480 CONTINUE COND3120
490 IF (MATZG(JG, I, K).EQ.100.OR.MATZG(JG, I, K).EQ.300) GO TO 530 COND3130
IF (ABS(T(I, J, K)-T(I, J+1, K)).LT.TDMIN) GO TO 500 COND3140
RAD = 0.1713E-8*(ZBBTH(JGL, I, K)**4 - ZBBTL(JG, I, K)**4)/(ZEMH( COND3150
' JGL, I, K) + ZEML(JG, I, K) - 1.0) COND3160
HX = CONZ(I, J, K)*(T(I, J + 1, K) - T(I, J, K) + CONZ(I, J + 1, KCOND3170
' )*RAD*DXT)/(T(I, J + 1, K) - T(I, J, K) - CONZ(I, J, K)*RAD) COND3180
GO TO 540 COND3190
500 MATZG(JG, I, K) = 300 COND3200
MATZG(JGL, I, K) = 300 COND3210

```



```

DO 510 JX=1,JM                                COND3220
IF (JGZ(JX).NE.JGL) GO TO 510                  COND3230
JPL = JX                                       COND3240
GO TO 520                                       COND3250
510 CONTINUE                                    COND3260
C RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE COND3270
C PRECEDING ITERATION AND PRESERVED IN THE ZEM=7BBT ARRAYS. COND3280
C CONL IS CONZ(I,JPL,K)                        COND3290
520 CONL=ZEM(L,JGL,I,K)                        COND3300
C CONL IS CONZ(I,JPL+1,K)                     COND3310
CONL=ZBBT(L,JGL,I,K)                          COND3320
HX=CONL                                         COND3330
DXT=(ZL(JPL)-ZF(JPL))                        COND3340
KZ(1,JPL,K)=DELZ(JPL)/(CONL*DXT+HX)          COND3350
530 HX = CONZ(I, J, K)                        COND3360
540 KZ(1,J,K)=DELZ(J)/(CONZ(I,J+1,K)*DXT+HX) COND3370
IF (.NOT.ZG(JG,I,K).NE.300) GO TO 550         COND3380
IF (NOMES) GO TO 550                          COND3390
IERROR(1) = IERROR(1) + 1                    COND3400
WRITE (N,100) ICOND(6),NNITER,ICOND(7),JGL,ICOND(8),JG,ICOND(9),I,COND3410
1ICOND(13),K                                   COND3420
CE *****COND3430
550 CONTINUE                                    COND3440
C HAVE ALL POINTS IN THE COLUMN BEEN CONSIDERED ? -'YES,NO'- COND3450
560 CONTINUE                                    COND3460
C HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'- COND3470
570 CONTINUE                                    COND3480
C HAVE ALL PLANES BEEN CONSIDERED ? -'YES,NO'- COND3490
580 CONTINUE                                    COND3500
CE *****COND3510
C =====COND3520
C =====COND3530
C CALCULATE THE AVERAGE THETA CONDUCTIVITIES BETWEEN POINTS. COND3540
C =====COND3550
C CALCULATE THE THETA CONDUCTIVITIES IN EACH THETA PLANE. COND3560
DO 640 K=1,KM                                  COND3570
K0=K61(K)                                       COND3580
C =====COND3590
C CALCULATE THE THETA CONDUCTIVITIES IN EACH AXIAL COLUMN IN THE COND3600
C PLANE.                                         COND3610
DO 630 I=2,IM                                  COND3620
C =====COND3630
C CALCULATE THE THETA CONDUCTIVITIES IN THE COLUMN. COND3640
DO 620 J=2,JM                                  COND3650
C DOES THE GRIDLINE HAVE ANYWHERE A GAP OR A COOLANT ADJACENT TO IT COND3660
C .? -'NO,YES'-                                COND3670

```

```

IF (KG,EQ.0) GO TO 600
=====COND3680
C IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS COND3690
C POINT ? --NO,YES,-- COND3700
C IF (GAPT(KG,I,J),EQ.0.0) GO TO 600 COND3710
C IS IT A GAP OR A COOLANT WHICH IS ADJACENT ? --COOLANT,GAP,-- COND3720
C IF (GAPT(KG,I,J),LT.0.0) GO TO 620 COND3730
=====COND3740
C CALCULATE THE AVERAGE CONDUCTIVITY WITH A GAP PRESENT COND3750
CB NI = MATIG(KG, I, J) COND3760
TH = TUBTH(KG, I, J) COND3770
TB = TPBTL(KG, I, J) COND3780
FTR = RP(I) COND3790
FIT = ZP(J) COND3800
FTT = TP(K) COND3810
GWID = GAPT(KG, I, J) COND3820
DXHS = TATIOH(KG) COND3830
C CONVERT GWID AND DXHS FROM RADIANS TO FEET FOR CYLINDRICAL COND3840
C GEOMETRY. COND3850
C IF (ISHAPE,NE.0) GO TO 590 COND3860
C GWID = GWID*RP(I) COND3870
DXHS = DXHS*RP(I) COND3880
C CALCULATE THE LOCAL GAS CONDUCTIVITY. COND3890
590 CALL MADATA COND3900
HX = TEMH(KG, I, J) + TEMPL(KG, I, J) - 1.0 COND3910
CX = .1713E-R*(TH**2 + TB**2)*(TH + TB)/BX COND3920
X1 = CONT(I, J, K)*TATIOH(KG, I, J) COND3930
X2 = 1./((GK/GWID+CX)*DXHS) COND3940
X3 = CONT(I, J, K + 1) COND3950
C PRESERVE THE DATA REQUIRED FOR CALCULATING THETA GAP BOUNDARY COND3960
C TEMPERATURES BY STORING IT IN THE TEM-TBTT LOCATIONS FOR THE COND3970
C CURRENT GAPLINE. COND3980
C TEMPL(KG, I, J) = CONT(I, J, K) COND3990
C TEMH(KG, I, J) = CONT(I, J, K + 1) COND4000
C TUBTL(KG, I, J) = BX COND4010
C TUBTH(KG, I, J) = GK COND4020
C KT(I,J,K) = 1./((X1+X2+X3)/TATIOK(KG)) COND4030
C GO TO 810 COND4040
CE *****COND4050
CE *****COND4060
C ARE THE CONDUCTIVITIES ON BOTH SIDES OF THE GRID PLANE IDENTICAL COND4070
C .? --NO,YES,-- COND4080
C 600 IF (CONT(I,J,K).NE.CONT(I,J,K+1)) GO TO 610 COND4090
C ASSIGN ONE OF THEM AS THE AVERAGE CONDUCTIVITY COND4100
C KT(I, J, K) = 1.0/CONT(I, J, K) COND4110
C GO TO 810 COND4120
COND4130

```

```

C      CALCULATE THE AVERAGE CONDUCTIVITY WHEN NO GAP OR COOLANT IS      COND4140
C      PRESENT                                                                COND4150
610 KI(I, J, K) = 1.0/(CONT(I, J, K + 1) + (CONT(I, J, K) - CONT(I, J    COND4160
    ,K+1))*TLN(K))                                                            COND4170
    GO TO 810                                                                    COND4180
CB     CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ADJACENT          COND4190
620 DZ=DELTA(K)                                                                COND4200
C      CONVERT DZ FROM RADIAN TO FEET FOR CYLINDRICAL GEOMETRY.          COND4210
    IF (ISHAPE.EQ.0) DZ=DZ*RP(I)                                              COND4220
C      IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GRIDPLANE?-'NO,YES'-  COND4230
    IF (GAP(I,KG,I,J),LT.(-1.5E-10)) GO TO 700                               COND4240
C      CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE HIGH     COND4250
C      INDEX SIDE.                                                            COND4260
    DXT=TL(K)-TP(K)                                                            COND4270
C      CONVERT DXT FROM RADIAN TO FEET FOR CYLINDRICAL GEOMETRY.          COND4280
    IF (ISHAPE.EQ.0) DXT=DXT*RP(I)                                            COND4290
C      EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.                 COND4300
    IF (KG.EQ.NTG) GO TO 680                                                  COND4310
C      FIND THE HIGH THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.        COND4320
    KG1=KG+1                                                                    COND4330
    DO 640 KK=KG1,NTG                                                         COND4340
    IF (GAP(KK,I,J),LT.(-1.5E-10)) GO TO 630                                  COND4350
    GO TO 640                                                                    COND4360
C      EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT      COND4370
C      THE HIGH THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.            COND4380
630 IF (GAP(KK,I,J),LT.(-2.5E-10)) GO TO 680                                COND4390
    KGH=KK                                                                      COND4400
    GO TO 650                                                                    COND4410
640 CONTINUE                                                                    COND4420
650 IF (ABS(TEMH(KG,I,J)).GT.,999E6.OR.ABS(TEML(KGH,I,J)).GT.,999E6) 6COND4430
    GO TO 660                                                                    COND4440
    IF (ABS(T(I,J,K)-T(I,J,K+1)),LT.TDMIN) GO TO 670                         COND4450
    RAD = 0.1713E-8*(TBBTH(KG, I, J)**4 - TBBTL(KGH, I, J)**4)/(TEMH(    COND4460
    * KG, I, J) + TEML(KGH, I, J) - 1.0)                                       COND4470
    HX = CONT(I, J, K + 1)*(T(I, J, K) - T(I, J, K + 1) - RAD*CONT(I,    COND4480
    * J, K)*DXT)/(T(I, J, K) - T(I, J, K + 1) + RAD*CONT(I, J, K + 1))  COND4490
    MATIG(KG, I, J) = 200                                                       COND4500
    MATIG(KGH, I, J) = 200                                                       COND4510
    GO TO 690                                                                    COND4520
660 MATIG(KG, I, J) = 100                                                       COND4530
    MATIG(KGH, I, J) = 100                                                       COND4540
    GO TO 680                                                                    COND4550
670 MATIG(KG, I, J) = 300                                                       COND4560
    MATIG(KGH, I, J) = 300                                                       COND4570
680 HX = CONT(I, J, K + 1)                                                       COND4580
690 KT(I, J, K) = DZ/(CONT(I, J, K)*DXT + HX)                                COND4590

```

	GO TO R10	COND4600
C	ARE TWO COOLANTS ADJACENT? -;NO,YES;-	COND4610
700	IF (GAPT(KG,I,J).GT.(-2.5E-10)) GO TO 710	COND4620
C	DO NOT ALLOW HEAT TRANSFER BETWEEN COOLANTS	COND4630
	KT(I, J, K) = 1.0E-10	COND4640
	GO TO R10	COND4650
C	CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE LOW	COND4660
C	INDEX SIDL.	COND4670
710	DXT=TP(K+1)-TL(K)	COND4680
C	CONVERT DXT FROM RADIANS TO FEET FOR CYLINDRICAL GEOMETRY.	COND4690
	IF (ISHAPE.EQ.0) DXT=DXT*RP(I)	COND4700
C	EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.	COND4710
	IF (KJ.EQ.1) GO TO 790	COND4720
C	FIND THE LOW THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.	COND4730
	KG1=KG-1	COND4740
	DO 730 KK=1,KG1	COND4750
	KG2=KG-KK	COND4760
	IF (GAPT(KG2,I,J).LT.(-.5E-10)) GO TO 720	COND4770
	GO TO 730	COND4780
C	EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT	COND4790
C	THE LOW THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.	COND4800
720	IF (GAPT(KG2,I,J).LT.(-2.5E-10)) GO TO 790	COND4810
	KGL=KG2	COND4820
	GO TO 740	COND4830
730	CONTINUE	COND4840
740	IF (MATIG(KG,I,J).EQ.100.OR.MATIG(KG,I,J).EQ.300) GO TO 790	COND4850
	IF (ABS(T(I,J,K)-T(I,J,K+1)).LT.TDMIN) GO TO 750	COND4860
	RAD = 0.1713E-8*(TBBTH(KGL, I, J)**4 - TBBTL(KG, I, J)**4)/(TEMP	COND4870
	* KGL, I, J) + TEMPL(KG, I, J) - 1.0)	COND4880
	HX = CONT(I, J, K)*(T(I, J, K + 1) - T(I, J, K) + CONT(I, J, K + 1	COND4890
	*)*RAD*DXT)/(T(I, J, K + 1) - T(I, J, K) - CONT(I, J, K)*RAD)	COND4900
	GO TO 800	COND4910
750	MATIG(KG, I, J) = 300	COND4920
	MATIG(KGL, I, J) = 300	COND4930
	DO 760 KX=1,KM	COND4940
	IF (KGT(KX).NE.KGL) GO TO 760	COND4950
	KPL=KX	COND4960
	GO TO 770	COND4970
760	CONTINUE	COND4980
C	RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE	COND4990
C	PRECEEDING ITERATION AND PRESERVED IN THE TEM-TBBT ARRAYS.	COND5000
C	CONL IS CONT(I,J,KPL)	COND5010
770	CONL=TEMP(KGL,I,J)	COND5020
C	CONH IS CONT(I,J,KPL+1)	COND5030
	CONH = TBBTL(KGL, I, J)	COND5040
	HX=CONH	COND5050

```

DZP=DELTA(KPL)
DXTP= TL(KPL)-TP(KPL)
CONVERT DZP AND DXTP FROM RADIANS TO FEET FOR CYLINDRICAL
GEOMETRY.
IF (ISHAPE.NE.0) GO TO 780
DZP=DZP*RP(I)
DXTP=DXTP*RP(I)
780 KT(1,J,KPL)=DZP/(CONL*DXTP+HX)
790 HX = CONT(I, J, K)
800 KT(1, J, K) = DZ/(CONT(I, J, K + 1)*DXT + HX)
IF (MATTG(KG,I,J).NE.300) GO TO A10
IF(NOMES) GO TO 810
IERROR(1) = IERROR(1) + 1
WRITE (6,100) ICOND(10),NNITER,ICOND(11),JGL,ICOND(12),JG,ICOND(13)
1,I,ICOND(4),J
CE *****
810 CONTINUE
C HAVE ALL POINTS IN THE COLUMN BEEN CONSIDERED ? -'YES,NO'-
820 CONTINUE
C HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'-
830 CONTINUE
C HAVE ALL PLANES BEEN CONSIDERED ? -'YES,NO'-
840 CONTINUE
CE *****
C =====
RETURN
END

```

```

COND5060
COND5070
COND5080
COND5090
COND5100
COND5110
COND5120
COND5130
COND5140
COND5150
COND5160
COND5170
COND5180
COND5190
COND5200
COND5210
COND5220
COND5230
COND5240
COND5250
COND5260
COND5270
COND5280
COND5290
COND5300
COND5310
COND5320

```

```

SUBROUTINE STEP(S,NOTEST) STEP 10
INCLUDE COMDIM STEP 20
C=====STEP 30
CB CALCULATE THE NEW TEMPERATURES AFTER ONE TIMESTEP STEP 40
C=====STEP 50
C=====STEP 60
LOGICAL NOTEST STEP 70
DIMENSION X (MQ),Y (MQ) STEP 80
C=====STEP 90
C/ PERFORM A RADIAL SET OF ITERATIONS STEP 100
DO 12 J= 2, JM STEP 110
DO 11 K= 2, KM STEP 120
C=====STEP 130
I=2 STEP 140
C1=KR(I-1,J,K)*KR(I-1,J,K) STEP 150
C7=KCP(I,J,K)/DT STEP 160
C2=KR(I,J,K)*KR(I,J,K)/2.0 STEP 170
E=-C7 STEP 180
X(I)=C2/E STEP 190
Y(I)=(-C7*T(I,J,K)+C1*(T(I-1,J,K)-T(I,J,K)))+C2* STEP 200
X(I(1+1,J,K)-T(I,J,K)-T(I,J,K))+RZ(I,J-1,K)*KZ(I,J-1,K)*(T(I,J-1,K) STEP 210
X-T(I,J,K))+RZ(I,J,K)*KZ(I,J,K)*(T(I,J+1,K)-T(I,J,K))+RT(I,J,K-1) STEP 220
X*KT(I,J,K-1)*(T(I,J,K-1)-T(I,J,K))+RT(I,J,K)*KT(I,J,K)*(T(I,J,K+1) STEP 230
X-T(I,J,K))+W(I,J,K)))/E STEP 240
C=====STEP 250
DO 10 I=3, IM1 STEP 260
C1=KR(I-1,J,K)*KR(I-1,J,K)/2.0 STEP 270
C7=KCP(I,J,K)/DT STEP 280
C2=KR(I,J,K)*KR(I,J,K)/2.0 STEP 290
E=-C7-C1*X(I-1) STEP 300
X(I)=C2/E STEP 310
Y(I)=(-C7*T(I,J,K)+C1*(T(I-1,J,K)-T(I,J,K)-T(I,J,K))+C2* STEP 320
X(T(1+1,J,K)-T(I,J,K)-T(I,J,K))+RZ(I,J-1,K)*KZ(I,J-1,K)*(T(I,J-1,K) STEP 330
X-T(I,J,K))+RZ(I,J,K)*KZ(I,J,K)*(T(I,J+1,K)-T(I,J,K))+RT(I,J,K-1) STEP 340
X*KT(I,J,K-1)*(T(I,J,K-1)-T(I,J,K))+RT(I,J,K)*KT(I,J,K)*(T(I,J,K+1) STEP 350
X-T(I,J,K))+W(I,J,K))-C1*X(I-1))/E STEP 360
10 CONTINUE STEP 370
C=====STEP 380
I= 1M STEP 390
C1=KR(I-1,J,K)*KR(I-1,J,K)/2.0 STEP 400
C7=KCP(I,J,K)/DT STEP 410
C2=KR(I,J,K)*KR(I,J,K) STEP 420
E=-C7-C1*X(I-1) STEP 430
X(I)=0.0 STEP 440
Y(I)=(-C7*T(I,J,K)+C1*(T(I-1,J,K)-T(I,J,K)-T(I,J,K))+C2* STEP 450

```

```

X(T(I+1,J,K)-T(I,J,K) )+R7(I,J-1,K)*KZ(I,J-1,K)*(T(I,J-1,K)STEP 460
X=T(I,J,K))+R7(I,J,K)*KZ(I,J,K)*(T(I,J+1,K)-T(I,J,K))+RT(I,J,K-1) STEP 470
X*KT(I,J,K-1)*(T(I,J,K-1)-T(I,J,K))+RT(I,J,K)*KT(I,J,K)*(T(I,J,K+1)STEP 480
X=T(I,J,K))+W(I,J,K))-C1*Y(I-1))/E STEP 490
C=====STEP 500
15 TT(I,J,K)=Y(I)-X(I)*TT(I+1,J,K) STEP 510
I=I-1 STEP 520
IF (I.GT.1) GO TO 15 STEP 530
C=====STEP 540
11 CONTINUE STEP 550
12 CONTINUE STEP 560
CE *****STEP 570
C=====STEP 580
C/ PERFORM A AXIAL SET OF ITERATIONS STEP 590
DO 22 I= 2,IM STEP 600
DO 21 K= 2,KM STEP 610
C=====STEP 620
DO 20 J=2,JM STEP 630
C3=K/(I,J-1,K)*KZ(I,J-1,K) STEP 640
IF (J .EQ. 2) C3 = 0.0 STEP 650
C7=KCP(I,J,K)/DT*2.0 STEP 660
C4=K/(I,J,K)*KZ(I,J,K) STEP 670
IF (J .EQ. JM) C4 = 0.0 STEP 680
E=-C7-C3*X(J-1) STEP 690
X(J)=C4/E STEP 700
Y(J)=(-C7*TT(I,J,K)+C3*T(I,J-1,K)+C4*T(I,J+1,K)-C3*Y(J-1))/E STEP 710
20 CONTINUE STEP 720
C=====STEP 730
J=JM STEP 740
25 TT(I,J,K)=Y(J)-X(J)*TT(I,J+1,K) STEP 750
J=J-1 STEP 760
IF (J.GT.1) GO TO 25 STEP 770
C=====STEP 780
21 CONTINUE STEP 790
22 CONTINUE STEP 800
CE STEP 810
C=====STEP 820
C/ PERFORM A THETA SET OF ITERATIONS STEP 830
DO 32 I= 2,IM STEP 840
DO 31 J= 2,JM STEP 850
C=====STEP 860
DO 30 K=2,KM STEP 870
C5=KT(I,J,K-1)*KT(I,J,K-1) STEP 880
IF (K .EQ. 2) C5 = 0.0 STEP 890
C7=KCP(I,J,K)/DT*2.0 STEP 900
C6=KT(I,J,K)*KT(I,J,K) STEP 910

```

```

      IF (K .EQ. KM) C6 = 0.0                      STEP 920
      E=-C7-C5*X(K-1)                             STEP 930
      X(K)=C6/E                                    STEP 940
      Y(K)=(-C7*TT(I,J,K)+C5*T(I,J,K-1)+C6*T(I,J,K+1)-C5*Y(K-1))/E STEP 950
30 CONTINUE                                       STEP 960
C=====STEP 970
      K= KM                                       STEP 980
35 T(I,J,K)=Y(K)-X(K)*T(I,J,K+1)                STEP 990
      K=K-1                                       STEP 1000
      IF (K.GT.1) GO TO 35                        STEP 1010
C=====STEP 1020
31 CONTINUE                                       STEP 1030
32 CONTINUE                                       STEP 1040
CE *****STEP 1050
      IF (.NOT.SW(10).OR.NOTEST) GO TO 50        STEP 1060
      DO 40 I=2,IM                                STEP 1070
      DO 40 J=2,JM                                STEP 1080
      DO 40 K=2,KM                                STEP 1090
      IF (T(I,J,K).LT.0.0.OR.T(I,J,K).GT.1.0E6) RETURN 1 STEP 1100
40 CONTINUE                                       STEP 1110
50 RETURN                                         STEP 1120
C=====STEP 1130
      END                                         STEP 1140

```



```

SUBROUTINE COOL                                COOL 10
INCLUDE      COMDIM                            COOL 20
=====COOL 30
C CALCULATE THE COOLANT TEMPERATURES          COOL 40
=====COOL 50
C ERROR MESSAGES                              COOL 60
C   COOL1 , COOL2 , COOL3                     COOL 70
C   THERE IS A PROGRAM LIMITATION WHICH MAY BE ENCOUNTERED COOL 80
C   WHEN CALCULATING COOLANT TEMPERATURES IN THE PRESENCE COOL 90
C   OF RADIATION BETWEEN COOLANT BLOCK BOUNDARIES. IT COOL 100
C   ARISES WHEN A COOLANT POINT AND AN ADJACENT MATERIAL COOL 110
C   POINT APPROACH THE SAME TEMPERATURE. THESE MESSAGES COOL 120
C   INDICATE THAT THIS LIMITATION HAS BEEN ENCOUNTERED COOL 130
C   AND CIRCUMVENTED BY ASSIGNING ZERO COOLANT HEAT COOL 140
C   TRANSFER AT THE LEVEL INDICATED. BY USING A SMALLER COOL 150
C   TIME STEP, THIS DIFFICULTY CAN USUALLY BE ELIMINATED. COOL 160
C                                               COOL 170
C   COOL1 HEAT TRANSFER WITHIN A RADIAL FLOW COOLANT IS COOL 180
C   ZERO IN BLOCK L AT RADIAL POINT LEVEL I. COOL 190
C   COOL2 HEAT TRANSFER WITHIN AN AXIAL FLOW COOLANT IS COOL 200
C   ZERO IN BLOCK L AT AXIAL POINT LEVEL J. COOL 210
C   COOL3 HEAT TRANSFER WITHIN A THETA FLOW COOLANT IS COOL 220
C   ZERO IN BLOCK L AT THETA POINT LEVEL K. COOL 230
C =====COOL 240
C LOGICAL ANTI, NOMES, IFP, IRAD, IXRAD COOL 250
C DIMENSION TC(MU), ICOOL(6) COOL 260
C =====COOL 270
C DATA(ICOOL(I), I=1,6) / 1H1,1H2,1H3,3H I=,3H J=,3H K= / COOL 280
100 FORMAT(14H ERROR AT COOL, A1, 17H ON ITERATION NO., I4, 6H AT L=, COOL 290
'13, A3, I2) COOL 300
C =====COOL 310
C =====COOL 320
C NNITER = NITER + 1 COOL 330
C NOMES=.FALSE. COOL 340
C IF(.NOT.SW(10)) GO TO 105 COOL 350
C IF(CS1.LT.1.0) NOMES=.TRUE. COOL 360
105 IF(FIRST) NOMES=.TRUE. COOL 370
C =====COOL 380
C CALCULATE THE COOLANT TEMPERATURES IN EACH BLOCK. COOL 390
C DO 650 L=1, LMAX COOL 400
C IS IT A COOLANT BLOCK ? -'NO, YES,- COOL 410
C IF (MB(L).GE.0) GO TO 650 COOL 420
C N = - MB(L) COOL 430
C TIN = TI(N) COOL 440
C DEFINE THE CORNERS OF THE COOLANT BLOCK COOL 450

```

	ILS = IL(L)	COOL 460
	JLS = JL(L)	COOL 470
	KLS=KL(L)	COOL 480
	IHS = IH(L)	COOL 490
	JHS = JH(L)	COOL 500
	KHS=KH(L)	COOL 510
C	FIND THE HIGH BOUNDING GAPLINES OF THE COOLANT BLOCK.	COOL 520
	IF (IHS.EQ.IMAX) GO TO 110	COOL 530
	IGHS = IGR(IHS)	COOL 540
	GO TO 120	COOL 550
110	IGHS = NRG	COOL 560
120	IF (JHS.EQ.JMAX) GO TO 130	COOL 570
	JGHS = JGZ(JHS)	COOL 580
	GO TO 140	COOL 590
130	JGHS = NZG	COOL 600
140	IF (KHS.EQ.KMAX) GO TO 150	COOL 610
	KGHS=KGZ(KHS)	COOL 620
	GO TO 160	COOL 630
150	KGHS=NTG	COOL 640
C	DETERMINE THE DIRECTION OF COOLANT FLOW -'RADIAL,AXIAL,THETA'-	COOL 650
160	ANTI = .FALSE.	COOL 660
	IF(IPATH(N).LT.0) ANTI=.TRUE.	COOL 670
	I = IABS(IPATH(N))	COOL 680
	GO TO (490,170,330), I	COOL 690
C	=====	COOL 700
C	CALCULATE THE AXIAL COOLANT TEMPERATURES.	COOL 710
C	=====	COOL 720
C	ASSIGN THE INLET TEMPERATURE	COOL 730
170	TC(JLS)=TIN	COOL 740
	IF(ANTI) TC(JHS)=TIN	COOL 750
C	CALCULATE THE COOLANT TEMPERATURES ALONG THE FLOW DIRECTION.	COOL 760
	DO 320 J1=JLS,JHS	COOL 770
	J=J1	COOL 780
	IF(ANTI) J=JHS+JLS-J	COOL 790
	SUMKA = 0.0	COOL 800
	SUMKAT = 0.0	COOL 810
	IFP = .FALSE.	COOL 820
	IXRAD = .FALSE.	COOL 830
C	CALCULATE THE HEAT TRANSFER ACROSS THE COOLANT BOUNDARIES AT THE	COOL 840
C	CURRENT AXIAL LEVEL.	COOL 850
C	SUM THE HEAT TRANSFER ACROSS THE RADIAL BOUNDARIES OF THE COOLANT.	COOL 860
	DO 190 K=KLS,KHS	COOL 870
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX	COOL 880
C	.SIDE? -'YES,NO'-	COOL 890
	IF (ILS.LE.2) GO TO 180	COOL 900
C	SUM THE HEAT TRANSFER ACROSS THE LOW RADIAL BOUNDARY.	COOL 910

	I = ILS - 1	COOL 920
	IRAD = .FALSE.	COOL 930
	IF(MATRG(IGHS,J,K).EQ.200) IRAD=.TRUE.	COOL 940
	IF(IRAD) IXRAD=.TRUE.	COOL 950
	X = RR(I, J, K)*KR(I, J, K)	COOL 960
	SUMKAT = SUMKAT + X*T(I, J, K)	COOL 970
	SUMKA = SUMKA + X	COOL 980
	IF(IRAD) CALL RANGE(IFP,T(I,J,K),THI,TLO)	COOL 990
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX	COOL,000
C	.SIDE? -'YES,NO'=-	COOL1010
C	180 IF (IHS.GE.IL) GO TO 190	COOL1020
C	SUM THE HEAT TRANSFER ACROSS THE HIGH RADIAL BOUNDARY.	COOL1030
	I = IHS	COOL1040
	IRAD = .FALSE.	COOL1050
	IF(MATRG(IGHS,J,K).EQ.200) IRAD=.TRUE.	COOL1060
	IF(IRAD) IXRAD=.TRUE.	COOL1070
	X = RR(I, J, K)*KR(I, J, K)	COOL1080
	SUMKAT = SUMKAT + X*T(I + 1, J, K)	COOL1090
	SUMKA = SUMKA + X	COOL1100
	IF(IRAD) CALL RANGE(IFP,T(I+1,J,K),THI,TLO)	COOL1110
C	HAS THE HEAT TRANSFER THROUGH ALL RADIAL POINTS BEEN	COOL1120
C	.CALCULATED? -'YES,NO'=-	COOL1130
C	190 CONTINUE	COOL1140
C	SUM THE HEAT TRANSFER ACROSS THE THETA BOUNDARIES OF THE COOLANT.	COOL1150
	DO 210 I=ILS,IHS	COOL1160
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX	COOL1170
C	.SIDE? -'YES,NO'=-	COOL1180
	IF (KLS.LE.2) GO TO 200	COOL1190
C	SUM THE HEAT TRANSFER ACROSS THE LOW THETA BOUNDARY.	COOL1200
	K=KLS-1	COOL1210
	IRAD = .FALSE.	COOL1220
	IF(MATRG(KGHS,I,J).EQ.200) IRAD=.TRUE.	COOL1230
	IF(IRAD) IXRAD=.TRUE.	COOL1240
	X = RT(I, J, K)*KT(I, J, K)	COOL1250
	SUMKAT = SUMKAT + X*T(I, J, K)	COOL1260
	SUMKA = SUMKA + X	COOL1270
	IF(IRAD) CALL RANGE(IFP,T(I,J,K),THI,TLO)	COOL1280
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX	COOL1290
C	.SIDE? -'YES,NO'=-	COOL1300
C	200 IF (KHS.GE.KM) GO TO 210	COOL1310
C	SUM THE HEAT TRANSFER ACROSS THE HIGH THETA BOUNDARY.	COOL1320
	K=KHS	COOL1330
	IRAD = .FALSE.	COOL1340
	IF(MATRG(KGHS,I,J).EQ.200) IRAD=.TRUE.	COOL1350
	IF(IRAD) IXRAD=.TRUE.	COOL1360
	X = RT(I, J, K)*KT(I, J, K)	COOL1370

	SUMKAT = SUMKAT + X*T(I, J, K + 1)	COOL1380
	SUMKA = SUMKA + X	COOL1390
	IF (IRAD) CALL RANGE(IFP,T(I,J,K+1),THI,TLO)	COOL1400
C	HAS THE HEAT TRANSFER THROUGH ALL THETA POINTS BEEN	COOL1410
C	CALCULATED ? -'YES,NO'-	COOL1420
210	CONTINUE	COOL1430
C	CALCULATE THE COOLANT TEMPERATURES FOR THE CURRENT AXIAL	COOL1440
C	INCREMENT.	COOL1450
	WCP = FLOW(N)*RCPC(N, J)	COOL1460
	EX = SUMKA/WCP	COOL1470
C	IS THE RADIATION EFFECT UPON THE HEAT TRANSFER COEFFICIENT TOO	COOL1480
C	LARGE? -'NO,YES'-	COOL1490
	IF (EX.GT.-30.) GO TO 220	COOL1500
	GO TO 270	COOL1510
220	EXPO = EXP(-EX)	COOL1520
C	CALCULATE THE OUTLET COOLANT TEMPERATURE FOR THE AXIAL INCREMENT.	COOL1530
	X = TC(J)*EXPO + SUMKAT/SUMKA*(1.0 - EXPO)	COOL1540
C	CALCULATE THE AVERAGE COOLANT TEMPERATURE FOR THE AXIAL INCREMENT.	COOL1550
230	TAV = (SUMKAT - WCP*(X - TC(J)))/SUMKA	COOL1560
C	DOES THE MEAN TEMPERATURE FALL BETWEEN THE INLET AND EXIT	COOL1570
C	TEMPERATURES ?	COOL1580
	IF (TC(J).GT.X) GO TO 240	COOL1590
	IF (TAV.GE.TC(J).AND.TAV.LE.X) GO TO 260	COOL1600
	GO TO 250	COOL1610
240	IF (TAV.GE.X.AND.TAV.LE.TC(J)) GO TO 260	COOL1620
250	TAV = (TC(J) + X)*0.5	COOL1630
C	DOES THE EXIT TEMPERATURE FALL WITHIN THE LIMITS IMPOSED BY THE	COOL1640
C	INLET TEMPERATURE AND THE ADJACENT MATERIAL TEMPERATURES ?	COOL1650
260	IF (.NOT.IXRAD) GO TO 280	COOL1660
	IF (TC(J).GT.THI) THI=TC(J)	COOL1670
	IF (TC(J).LT.TLO) TLO=TC(J)	COOL1680
	IF (X.LE.THI.AND.X.GE.TLO) GO TO 280	COOL1690
270	X = TC(J)	COOL1700
	TAV = TC(J)	COOL1710
	IF (NOMES) GO TO 280	COOL1720
	IERROR(1) = IERROR(1) + 1	COOL1730
	WHILE (6,100) ICOOL(2),NNITER,L,ICOOL(5),J	COOL1740
280	IF (ANT1) GO TO 290	COOL1750
	TC(J+1)=X	COOL1760
	GO TO 300	COOL1770
290	TC(J-1)=X	COOL1780
C	RECORD THE COOLANT TEMPERATURE THROUGHOUT THE CURRENT	COOL1790
C	AXIAL INCREMENT.	COOL1800
300	DO 310 I=ILS,IHS	COOL1810
	DO 310 K=KLS,KHS	COOL1820
	T(I, J, K) = TAV	COOL1830

```

310 CONTINUE
C HAVE ALL AXIAL LEVELS BEEN CALCULATED ? -'YES,NO'-
C 320 CONTINUE
C ASSIGN THE COOLANT OUTLET TEMPERATURE.
TOUT = X
GO TO 650
C =====
C CALCULATE THE THETA COOLANT TEMPERATURES,
C =====
C ASSIGN THE INLET TEMPERATURE
330 TIC(KLS)=TIN
IF (NITI) TIC(KHS)=TIN
C CALCULATE THE COOLANT TEMPERATURES ALONG THE FLOW DIRECTION.
DO 480 K1=KLS,KHS
K=K1
IF (NITI) K=KHS+KLS-K
SUMKA = 0.0
SUMKAT = 0.0
IFP = .FALSE.
IXRAD = .FALSE.
C CALCULATE THE HEAT TRANSFER ACROSS THE COOLANT BOUNDARIES
C AT THE CURRENT THETA LEVEL.
C SUM THE HEAT TRANSFER ACROSS THE RADIAL BOUNDARIES OF THE COOLANT.
DO 350 J=JLS,JHS
C IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX
C SIDE ? -'YES,NO'-
IF (ILS.LE.2) GO TO 340
C SUM THE HEAT TRANSFER ACROSS THE LOW RADIAL BOUNDARY.
I = ILS - 1
IRAD = .FALSE.
IF (MATRG(IGHS,J,K).EQ.200) IRAD=.TRUE.
IF (IRAD) IXRAD=.TRUE.
X = KR(I, J, K)*KR(I, J, K)
SUMKAT = SUMKAT + X*T(I, J, K)
SUMKA = SUMKA + X
IF (IRAD) CALL RANGE(IFP,T(I,J,K),THI,TLO)
C IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX
C SIDE ? -'YES,NO'-
340 IF (IHS.GE.IA) GO TO 350
C SUM THE HEAT TRANSFER ACROSS THE HIGH RADIAL BOUNDARY.
I = IHS
IRAD = .FALSE.
IF (MATRG(IGHS,J,K).EQ.200) IRAD=.TRUE.
IF (IRAD) IXRAD=.TRUE.
X = KR(I, J, K)*KR(I, J, K)
SUMKAT = SUMKAT + X*T(I + 1, J, K)

```

```

COOL1840
COOL1850
COOL1860
COOL1870
COOL1880
COOL1890
COOL1900
COOL1910
COOL1920
COOL1930
COOL1940
COOL1950
COOL1960
COOL1970
COOL1980
COOL1990
COOL2000
COOL2010
COOL2020
COOL2030
COOL2040
COOL2050
COOL2060
COOL2070
COOL2080
COOL2090
COOL2100
COOL2110
COOL2120
COOL2130
COOL2140
COOL2150
COOL2160
COOL2170
COOL2180
COOL2190
COOL2200
COOL2210
COOL2220
COOL2230
COOL2240
COOL2250
COOL2260
COOL2270
COOL2280
COOL2290

```

```

SUMKA = SUMKA + X                                COOL2300
IF (IRAD) CALL RANGE(IFP,T(I+1,J,K),THI,TLO)      COOL2310
C HAS THE HEAT TRANSFER THROUGH ALL RADIAL POINTS BEEN COOL2320
C .CALCULATED? -'YES,NO'-                          COOL2330
350 CONTINUE                                       COOL2340
C SUM THE HEAT TRANSFER ACROSS THE AXIAL BOUNDARIES OF THE COOLANT. COOL2350
DO 370 I=ILS,IHS                                  COOL2360
C IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX COOL2370
C .SIDE? -'YES,NO'-                                COOL2380
IF (JLS.LE.2) GO TO 360                            COOL2390
C SUM THE HEAT TRANSFER ACROSS THE LOW AXIAL BOUNDARY. COOL2400
J = JLS - 1                                         COOL2410
IRAD = .FALSE.                                      COOL2420
IF (MATZG(JGHS,I,K).EQ.200) IRAD=.TRUE.            COOL2430
IF (IRAD) IXRAD=.TRUE.                             COOL2440
X = KZ(I, J, K)*KZ(I, J, K)                        COOL2450
SUMKAT = SUMKAT + X*T(I, J, K)                     COOL2460
SUMKA = SUMKA + X                                    COOL2470
IF (IRAD) CALL RANGE(IFP,T(I,J,K),THI,TLO)        COOL2480
C IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX COOL2490
C .SIDE? -'YES,NO'-                                COOL2500
360 IF (JHS.GE.J+1) GO TO 370                       COOL2510
C SUM THE HEAT TRANSFER ACROSS THE HIGH AXIAL BOUNDARY. COOL2520
J = JHS                                              COOL2530
IRAD = .FALSE.                                      COOL2540
IF (MATZG(JGHS,I,K).EQ.200) IRAD=.TRUE.            COOL2550
IF (IRAD) IXRAD=.TRUE.                             COOL2560
X = KZ(I, J, K)*KZ(I, J, K)                        COOL2570
SUMKAT = SUMKAT + X*T(I, J + 1, K)                 COOL2580
SUMKA = SUMKA + X                                    COOL2590
IF (IRAD) CALL RANGE(IFP,T(I,J+1,K),THI,TLO)      COOL2600
C HAS THE HEAT TRANSFER THROUGH ALL AXIAL POINTS BEEN COOL2610
C .CALCULATED? -'YES,NO'-                          COOL2620
370 CONTINUE                                       COOL2630
C CALCULATE THE COOLANT TEMPERATURES FOR THE CURRENT THETA INCREMENT COOL2640
WCP = FLOW(N)*RCPC(N, K)                            COOL2650
EX = SUMKA/WCP                                       COOL2660
C IS THE RADIATION EFFECT UPON THE HEAT TRANSFER COEFFICIENT TOO COOL2670
C .LARGE? -'NO,YES'-                                COOL2680
IF (LX.GT.-30.) GO TO 380                           COOL2690
GO TO 430                                            COOL2700
380 EXPO = EXP(- EX)                                 COOL2710
C CALCULATE THE OUTLET COOLANT TEMPERATURE FOR THE THETA INCREMENT. COOL2720
X = TC(K)*EXPO + SUMKAT/SUMKA*(1.0 - EXPO)        COOL2730
C CALCULATE THE AVERAGE COOLANT TEMPERATURE FOR THE THETA INCREMENT. COOL2740
390 TAV = (SUMKAT - WCP*(X - TC(K)))/SUMKA        COOL2750

```

```

C      DOES THE MEAN TEMPERATURE FALL BETWEEN THE INLET AND EXIT      COOL2760
C      TEMPERATURES ?
C      IF (TC(K).GT.X) GO TO 400
C      IF (TAV.GE.TC(K).AND.TAV.LE.X) GO TO 420
C      GO TO 410
400  IF (TAV.GE.X.AND.TAV.LE.TC(K)) GO TO 420
410  TAV = (TC(K) + X)*0.5
C      DOES THE EXIT TEMPERATURE FALL WITHIN THE LIMITS IMPOSED BY THE COOL2830
C      INLET TEMPERATURE AND THE ADJACENT MATERIAL TEMPERATURES ?
C      420 IF (.NOT.IXRAD) GO TO 440
C      IF (TC(K).GT.THI) THI=TC(K)
C      IF (TC(K).LT.TLO) TLO=TC(K)
C      IF (X.LE.THI.AND.X.GE.TLO) GO TO 440
430  X = TC(K)
C      TAV = TC(K)
C      IF (NOMES) GO TO 440
C      IERROR(1) = IERROR(1) + 1
C      WRITE (6,110) ICOOL(3),NNITER,L,ICOOL(6),K
440  IF (ANTI) GO TO 450
C      TC(K+1)=X
C      GO TO 460
450  TC(K-1)=X
C      RECORD THE COOLANT TEMPERATURE THROUGHOUT THE CURRENT
C      THETA INCREMENT.
C      460 DO 470 I=ILS,IHS
C      DO 470 J=JLS,JHS
C      T(I, J, K) = TAV
470  CONTINUE
C      HAVE ALL THETA LEVELS BEEN CALCULATED ? -'YES,NO'-
C      480 CONTINUE
C      ASSIGN THE COOLANT OUTLET TEMPERATURE.
C      TO(N) = X
C      GO TO 650
C      =====
C      CALCULATE THE RADIAL COOLANT TEMPERATURES.
C      =====
C      ASSIGN THE INLET TEMPERATURE
C      490 TC(ILS)=TIN
C      IF (ANTI) TC(IHS)=TIN
C      CALCULATE THE COOLANT TEMPERATURES ALONG THE FLOW DIRECTION.
C      DO 540 I1=ILS,IHS
C      I=I1
C      IF (ANTI) I=IHS+ILS-I
C      SUMKA = 0.0
C      SUMKAT = 0.0
C      IFP = .FALSE.

```

	IXRAD = .FALSE.	COOL3220
C	CALCULATE THE HEAT TRANSFER ACROSS THE COOLANT BOUNDARIES AT THE	COOL3230
C	CURRENT RADIAL LEVEL.	COOL3240
C	SUM THE HEAT TRANSFER ACROSS THE AXIAL BOUNDARIES OF THE COOLANT.	COOL3250
	DO 510 K=KLS,KHS	COOL3260
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX	COOL3270
C	.SIDE? -'YES,NO'-	COOL3280
	IF (KLS.LE.2) GO TO 500	COOL3290
C	SUM THE HEAT TRANSFER ACROSS THE LOW AXIAL BOUNDARY.	COOL3300
	J=JLS-1	COOL3310
	IXRAD = .FALSE.	COOL3320
	IF (MATZ6(JGHS,I,K).EQ.200) IRAD=.TRUE.	COOL3330
	IF (IRAD) IXRAD=.TRUE.	COOL3340
	X = KZ(I, J, K)*KZ(I, J, K)	COOL3350
	SUMKAT = SUMKAT + X*T(I, J, K)	COOL3360
	SUMKA = SUMKA + X	COOL3370
	IF (IRAD) CALL RANGE(IFP,T(I,J,K),THI,TLO)	COOL3380
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX	COOL3390
C	.SIDE? -'YES,NO'-	COOL3400
	500 IF (JHS.GE.JM) GO TO 510	COOL3410
C	SUM THE HEAT TRANSFER ACROSS THE HIGH AXIAL BOUNDARY.	COOL3420
	J=JHS	COOL3430
	IXRAD = .FALSE.	COOL3440
	IF (MATZ6(JGHS,I,K).EQ.200) IRAD=.TRUE.	COOL3450
	IF (IRAD) IXRAD=.TRUE.	COOL3460
	X = KZ(I, J, K)*KZ(I, J, K)	COOL3470
	SUMKAT = SUMKAT + X*T(I, J + 1, K)	COOL3480
	SUMKA = SUMKA + X	COOL3490
	IF (IRAD) CALL RANGE(IFP,T(I,J+1,K),THI,TLO)	COOL3500
C	HAS THE HEAT TRANSFER THROUGH ALL AXIAL POINTS BEEN	COOL3510
C	.CALCULATED? -'YES,NO'-	COOL3520
	510 CONTINUE	COOL3530
C	SUM THE HEAT TRANSFER ACROSS THE THETA BOUNDARIES OF THE COOLANT.	COOL3540
	DO 530 J=JLS,JHS	COOL3550
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX	COOL3560
C	.SIDE? -'YES,NO'-	COOL3570
	IF (KLS.LE.2) GO TO 520	COOL3580
C	SUM THE HEAT TRANSFER ACROSS THE LOW THETA BOUNDARY.	COOL3590
	K=KLS-1	COOL3600
	IXRAD = .FALSE.	COOL3610
	IF (MATZ6(KGHS,I,J).EQ.200) IRAD=.TRUE.	COOL3620
	IF (IRAD) IXRAD=.TRUE.	COOL3630
	X = RT(I, J, K)*KT(I, J, K)	COOL3640
	SUMKAT = SUMKAT + X*T(I, J, K)	COOL3650
	SUMKA = SUMKA + X	COOL3660
	IF (IRAD) CALL RANGE(IFP,T(I,J,K),THI,TLO)	COOL3670

C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX	COOL3680
C	.SIDL? -'YES,NO'-	COOL3690
520	IF (KHS.GE.KM) GO TO 530	COOL3700
C	SUM THE HEAT TRANSFER ACROSS THE HIGH THETA BOUNDARY.	COOL3710
	K=KHS	COOL3720
	IRAD = .FALSE.	COOL3730
	IF (.NOT.(KHS.I, J).EQ.200) IRAD=.TRUE.	COOL3740
	IF (IRAD) IXRAD=.TRUE.	COOL3750
	X = RT(I, J, K)*KT(I, J, K)	COOL3760
	SUMKAT = SUMKAT + X*(I, J, K + 1)	COOL3770
	SUMKA = SUMKA + X	COOL3780
	IF (IRAD) CALL RANGE(IFP, T(I, J, K+1), THI, TLO)	COOL3790
C	HAS THE HEAT TRANSFER THROUGH ALL THETA POINTS BEEN	COOL3800
C	CALCULATED? -'YES,NO'-	COOL3810
530	CONTINUE	COOL3820
C	CALCULATE THE COOLANT TEMPERATURES FOR THE CURRENT	COOL3830
C	RADIAL INCREMENT.	COOL3840
	WCP = FLOW(N)*RCPC(N, I)	COOL3850
	EX = SUMKA/WCP	COOL3860
C	IS THE RADIATION EFFECT UPON THE HEATTRANSFER COEFFICIENT TOO	COOL3870
C	.LARGE? -'NO,YES'-	COOL3880
	IF (EX.GT.-30.) GO TO 540	COOL3890
	GO TO 590	COOL3900
540	EXPO = EXP(- EX)	COOL3910
C	CALCULATE THE OUTLET COOLANT TEMPERATURE FOR THE RADIAL INCREMENT.	COOL3920
	X = TC(I)*EXPO + SUMKAT/SUMKA*(1.0 - EXPO)	COOL3930
C	CALCULATE THE AVERAGE COOLANT TEMPERATURE FOR THE RADIAL INCREMENT	COOL3940
550	TAV = (SUMKAT - WCP*(X - TC(I)))/SUMKA	COOL3950
C	DOES THE MEAN TEMPERATURE FALL BETWEEN THE INLET AND EXIT	COOL3960
C	TEMPERATURES?	COOL3970
	IF (TC(I).GT.X) GO TO 560	COOL3980
	IF (TAV.GE.TC(I).AND.TAV.LE.X) GO TO 580	COOL3990
	GO TO 570	COOL4000
560	IF (TAV.GE.X.AND.TAV.LE.TC(I)) GO TO 580	COOL4010
570	TAV = (TC(I) + X)*0.5	COOL4020
C	DOES THE EXIT TEMPERATURE FALL WITHIN THE LIMITS IMPOSED BY THE	COOL4030
C	INLET TEMPERATURE AND THE ADJACENT MATERIAL TEMPERATURES?	COOL4040
580	IF (.NOT.IXRAD) GO TO 600	COOL4050
	IF (TC(I).GT.THI) THI=TC(I)	COOL4060
	IF (TC(I).LT.TLO) TLO=TC(I)	COOL4070
	IF (X.LE.THI.AND.X.GE.TLO) GO TO 600	COOL4080
590	X = TC(I)	COOL4090
	TAV = TC(I)	COOL4100
	IF (NOMES) GO TO 600	COOL4110
	IERROR(1) = IERROR(1) + 1	COOL4120
	WRITE (6,100) ICOOL(3),NNITER,L,ICOOL(4),I	COOL4130

```

600 IF (ANTI) GO TO 610                                COOL4140
    TC(I+1)=X                                          COOL4150
    GO TO 620                                          COOL4160
610 TC(I-1)=X                                          COOL4170
C   RECORD THE COOLANT TEMPERATURE THROUGHOUT THE CURRENT COOL4180
C   RADIAL INCREMENT.                                COOL4190
620 DO 630 J=JLS,JHS                                  COOL4200
    DO 630 K=KLS,KHS                                  COOL4210
    T(I, J, K) = TAV                                  COOL4220
630 CONTINUE                                          COOL4230
C   HAVE ALL RADIAL LEVELS BEEN CALCULATED ? -'YES,NO'- COOL4240
640 CONTINUE                                          COOL4250
C   ASSIGN THE COOLANT OUTLET TEMPERATURE.           COOL4260
    TO(I) = X                                          COOL4270
C   =====COOL4280
C   HAVE ALL BLOCKS BEEN CHECKED -'YES,NO'-         COOL4290
650 CONTINUE                                          COOL4300
    RETURN                                            COOL4310
C   =====COOL4320
C   *****COOL4330
C   *****COOL4340
C   *****COOL4350
C   =====COOL4360
C   SUBROUTINE RANGE(IFP, TX, THI, TLO)              COOL4370
C   =====COOL4380
C   DETERMINE THE CURRENT VALUES OF THE HIGHEST AND LOWEST ADJACENT COOL4390
C   MATERIAL POINT TEMPERATURES.                   COOL4400
C   =====COOL4410
C   LOGICAL IFP                                       COOL4420
C   =====COOL4430
C   IF (IFP) GO TO 100                                COOL4440
    THI = TX                                          COOL4450
    TLO = TX                                          COOL4460
    IFP = .TRUE.                                       COOL4470
    GO TO 110                                          COOL4480
100 IF (IX.GT.THI) THI=TX                              COOL4490
    IF (IX.LT.TLO) TLO=TX                              COOL4500
C   =====COOL4510
110 RETURN                                            COOL4520
    END                                              COOL4530

```

```

SUBROUTINE SURT                                     SURT  10
INCLUDE      COMDIM                                 SURT  20
C= =====SURT  30
C CALCULATE THE BOUNDARY TEMPERATURES ASSOCIATED WITH THE POINT SURT  40
C TEMPERATURE RESULTS OF THE CURRENT ITERATION. SURT  50
C= =====SURT  60
C ERROR STOPS= SURT  70
C   SURT  1  INSTABILITY ENCOUNTERED WHILE CALCULATING THE SURT  80
C             RADIAL BOUNDARY TEMPERATURES OF A GAP. USE A SURT  90
C             SMALLER TIME STEP. SURT 100
C   SURT  2  INSTABILITY ENCOUNTERED WHILE CALCULATING THE SURT 110
C             AXIAL BOUNDARY TEMPERATURES OF A GAP. USE A SURT 120
C             USE A SMALLER TIME STEP. SURT 130
C   SURT  3  INSTABILITY ENCOUNTERED WHILE CALCULATING THE SURT 140
C             THETA BOUNDARY TEMPERATURES OF A GAP. USE A SURT 150
C             SMALLER TIME STEP. SURT 160
C =====SURT 170
C DIMENSION      ISURT(4) SURT 180
C DATA(ISURT(I),I=1,4) /6HSURT , 1H1, 1H2, 1H3/ SURT 190
C =====SURT 200
C CALCULATE THE BOUNDARY TEMPERATURES ADJACENT GAPS AND COOLANTS, SURT 210
C =====SURT 220
C CALCULATE THE RADIAL BOUNDARY TEMPERATURES IN EACH RADIAL PLANE. SURT 230
C DO 240 I=1,IM SURT 240
C   IG=IOR(I) SURT 250
C   EXCLUDE THE NON-GAPLINES. SURT 260
C   IF (IG.EQ.0) GO TO 240 SURT 270
C =====SURT 280
C CALCULATE THE RADIAL BOUNDARY TEMPERATURES FOR EACH AXIAL SURT 290
C COLUMN IN THE PLANE. SURT 300
C DO 270 J=2,JM SURT 310
C =====SURT 320
C CALCULATE THE RADIAL BOUNDARY TEMPERATURES IN THE COLUMN. SURT 330
C DO 280 K=2,KM SURT 340
C IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS SURT 350
C POINT ? -NO,YES,- SURT 360
C IF (GAPR(IG,J,K).EQ.0.0) GO TO 240 SURT 370
C IS IT A COOLANT OR A GAP WHICH IS ADJACENT ? -COOLANT,GAP,- SURT 380
C IF (GAPR(IG,J,K).LT.0.0) GO TO 130 SURT 390
C =====SURT 400
C CALCULATE THE BOUNDARY TEMPERATURES WHEN A GAP IS ADJACENT. SURT 410
C RECOVER PARAMLTERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE SURT 420
C PRECEDING ITERATION AND PRESERVED IN THE REM-RBBT ARRAYS. SURT 430
C BX IS (REMH(IG,J,K)+REML(IG,J,K)-1.0). SURT 440
C BX = RBBTL(IG, J, K) SURT 450

```

C	GK IS THE GAS THERMAL CONDUCTIVITY.	SURT 460
	GK = RBBTH(IG, J, K)	SURT 470
C	CONL IS CONR(I,J,K)	SURT 480
	CONL = REML(IG, J, K)	SURT 490
C	CONH IS CONR(I+1,J,K)	SURT 500
	CONH = REMH(IG, J, K)	SURT 510
C	GET READY TO CALCULATE THE BOUNDARY TEMPERATURES OF THE GAP.	SURT 520
	B = RATIOB(IG, J, K)*CONL/CONH	SURT 530
	C = T(I, J, K) + B*T(I + 1, J, K)	SURT 540
	RGAP = 0.1713E-08*GAPR(IG, J, K)/RX	SURT 550
	ED = RATIOH(IG)*CONH/GAPR(IG, J, K)	SURT 560
C	ALLOW 10 ITERATIONS TO CALCULATE THE BOUNDARY TEMPERATURES.	SURT 570
	DO 120 KK=1,10	SURT 580
	IF (KK.EQ.1) GO TO 100	SURT 590
	TH = RBBTH(IG, J, K)	SURT 600
	TB = RBBTL(IG, J, K)	SURT 610
	GO TO 110	SURT 620
100	TH = T(I, J, K)	SURT 630
	TB = T(I + 1, J, K)	SURT 640
110	GH=(GK+RGAP*(TH**2+TB**2))*(TH+TB)*ED	SURT 650
	X1 = GH*C	SURT 660
	X2 = 1.0 + GH*(B + 1.0)	SURT 670
	RBBTL(IG, J, K) = (X1 + T(I + 1, J, K))/X2	SURT 680
	RBBTH(IG, J, K) = (X1 + T(I, J, K))/X2	SURT 690
C	ARE THE BOUNDARY TEMPERATURES CONVERGED ? -'YES,NO'=-	SURT 700
	IF (ABS(RBBTH(IG,J,K)-TH).LE.1.0) GO TO 260	SURT 710
C	HAVE THE 10 ITERATIONS BEEN PERFORMED ? -'YES,NO'=-	SURT 720
120	CONTINUE	SURT 730
C	NEGATIVE TEMPERATURES ARE PRESENT AND RESULT IN LOCAL	SURT 740
C	INSTABILITIES. TRY A SMALLER TIME STEP.	SURT 750
	IERROK(1) = ISURT(1)	SURT 760
	IERROK(2) = ISURT(2)	SURT 770
	CALL ERROR?	SURT 780
C	=====	SURT 790
C	CALCULATE THE BOUNDARY TEMPERATURES WHEN A COOLANT IS ADJACENT.	SURT 800
C	IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE ? -'NO,YES'=-	SURT 810
130	IF (GAPR(IG,J,K).LT.(-1.5E-10)) GO TO 190	SURT 820
C	CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE HIGH	SURT 830
C	INDEX SIDE.	SURT 840
C	RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE	SURT 850
C	PRECEEDING ITERATION AND PRESERVED IN THE REM=RBBT ARRAYS.	SURT 860
C	CONL IS CONR(I,J,K)	SURT 870
	CONL = REML(IG, J, K)	SURT 880
C	CONH IS CONR(I+1,J,K)	SURT 890
	CONH = RBBTL(IG, J, K)	SURT 900
C	EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.	SURT 910

	IF (IG.EQ.NRG.OR.MATRG(IG,J,K).NE.200) GO TO 170	SURT 920
C	FIND THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT 930
	IG1 = IG + 1	SURT 940
	DO 150 II=IG1,NRG	SURT 950
	IF (GAPR(II,J,K).LT.(-1.5E-10)) GO TO 140	SURT 960
	GO TO 150	SURT 970
C	EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT	SURT 980
C	THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT 990
140	IF (GAPR(II,J,K).LT.(-2.5E-10)) GO TO 170	SURT 1000
	IGH = II	SURT 1010
	GO TO 160	SURT 1020
150	CONTINUE	SURT 1030
160	RAD = 0.1713E-8*(RBBTH(IG, J, K)**4 - RBBTL(IGH, J, K)**4)/(REMH(SURT 1040
	IG, J, K) + REML(IGH, J, K) - 1.0)	SURT 1050
C	PRESERVE RAD IN THE REML ARRAY.	SURT 1060
	REML(IG, J, K) = RAD	SURT 1070
	HX = CONH*(T(I, J, K) - T(I + 1, J, K) - RAD*CONL*DELR(I)*RLN(I))/	SURT 1080
	(T(I, J, K) - T(I + 1, J, K) + RAD*CONH)	SURT 1090
	GO TO 180	SURT 1100
170	HX = CONH	SURT 1110
C	CALCULATE NEW BOUNDARY TEMPERATURES.	SURT 1120
180	RBBTH(IG, J, K) = (CONL*T(I + 1, J, K) + HX/(DELR(I)*RLN(I))*T(I,	SURT 1130
	J, K))/(CONL + HX/(DELR(I)*RLN(I)))	SURT 1140
	RBBTL(IG, J, K) = RBBTH(IG, J, K)	SURT 1150
	GO TO 200	SURT 1160
C	ARE TWO COOLANTS ADJACENT? -'NO,YES'-	SURT 1170
190	IF (GAPR(IG,J,K).GT.(-2.5E-10)) GO TO 200	SURT 1180
C	DO NOT ATTEMPT TO CALCULATE BOUNDARY TEMPERATURES.	SURT 1190
	GO TO 200	SURT 1200
C	CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE LOW	SURT 1210
C	INDEX SIDE.	SURT 1220
C	RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE	SURT 1230
C	PRECEDING ITERATION AND PRESERVED IN THE REM-RBBT ARRAYS.	SURT 1240
C	CONL IS CONR(I,J,K)	SURT 1250
200	CONL=RBBTH(IG,J,K)	SURT 1260
C	CONH IS CONR(I+1,J,K)	SURT 1270
	CONH=REMH(IG,J,K)	SURT 1280
C	EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.	SURT 1290
	IF (IG.EQ.1) GO TO 240	SURT 1300
	IF (MATRG(IG,J,K).NE.200) GO TO 240	SURT 1310
C	FIND THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT 1320
	IG1 = IG - 1	SURT 1330
	DO 220 II=1,IG1	SURT 1340
	IG2 = IG - II	SURT 1350
	IF (GAPR(IG2,J,K).LT.(-.5E-10)) GO TO 210	SURT 1360
	GO TO 220	SURT 1370

```

C      EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT          SURT1380
C      THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.                SURT1390
210  IF (GAPR(IG2,J,K).LT.(-2.5E-10)) GO TO 240                             SURT1400
      IGL = IG2                                                                SURT1410
      GO TO 230                                                                SURT1420
220  CONTINUE                                                                  SURT1430
C      RECOVER VALUE OF RAD STORED IN THE REML ARRAY.                        SURT1440
230  RAD = REML(IGL, J, K)                                                    SURT1450
      HX = CONL*(T(I + 1, J, K) - T(I, J, K) + CONH*RAD*RATIOB(IG, J, K) SURT1460
      *DELK(1)/RATIOK(IG))/(T(I + 1, J, K) - T(I, J, K) - CONL*RAD* SURT1470
      *RATIOB(IG, J, K))                                                       SURT1480
      GO TO 250                                                                SURT1490
240  HX=CONL                                                                    SURT1500
C      CALCULATE NEW BOUNDARY TEMPERATURES.                                  SURT1510
250  RUBTL(IG, J, K) = (CONH*T(I, J, K) + HX*RATIOK(IG)/DELR(I))*T(I + 1 SURT1520
      , J, K)/(CONH + HX*RATIOK(IG)/DELR(I))                                  SURT1530
      RUBTH(IG, J, K) = RUBTL(IG, J, K)                                       SURT1540
C      =====SURT1550
C      HAVE ALL POINTS IN THE COLUMN BEEN CONSIDERED ? -'YES,NO'-          SURT1560
260  CONTINUE                                                                    SURT1570
C      =====SURT1580
C      HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'-          SURT1590
270  CONTINUE                                                                    SURT1600
C      =====SURT1610
C      HAVE ALL RADIAL BOUNDARY TEMPERATURES BEEN CALCULATED ? -'YES,NO'- SURT1620
280  CONTINUE                                                                    SURT1630
C      =====SURT1640
C      =====SURT1650
C      CALCULATE THE AXIAL BOUNDARY TEMPERATURES IN EACH AXIAL PLANE.      SURT1660
      DO 470 J=1,JM                                                            SURT1670
      JG = JGZ(J)                                                              SURT1680
C      =====SURT1690
C      EXCLUDE THE NON-GAPLINES.                                             SURT1700
      IF (JG.EQ.0) GO TO 470                                                  SURT1710
C      =====SURT1720
C      CALCULATE THE AXIAL BOUNDARY TEMPERATURES FOR EACH RADIAL COLUMN SURT1730
C      COLUMN IN THE PLANE.                                                  SURT1740
      DO 480 I=2,IM                                                            SURT1750
C      =====SURT1760
C      CALCULATE THE AXIAL BOUNDARY TEMPERATURES IN THE COLUMN.            SURT1770
      DO 450 K=2,KM                                                            SURT1780
C      IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS SURT1790
C      POINT ? -'NO,YES'-                                                    SURT1800
      IF (GAPZ(JG,I,K).EQ.0.0) GO TO 450                                       SURT1810
C      IS IT A COOLANT OR A GAP WHICH IS ADJACENT ? -'COOLANT,GAP'- SURT1820
      IF (GAPZ(JG,I,K).LT.0.0) GO TO 320                                       SURT1830

```

```

C =====SURT1840
C CALCULATE THE BOUNDARY TEMPERATURES WHEN A GAP IS ADJACENT. SURT1850
C RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE SURT1860
C PRECEDING ITERATION AND PRESERVED IN THE ZEM-ZBBT ARRAYS. SURT1870
C BX IS (ZEMH(JG,I,K)+ZEML(JG,I,K)-1.0) SURT1880
C BX = ZBBTL(JG, I, K) SURT1890
C GK IS THE GAS THERMAL CONDUCTIVITY. SURT1900
C GK = ZBBTH(JG, I, K) SURT1910
C CONL IS CO.Z(I,J,K) SURT1920
C CONL = ZEML(JG, I, K) SURT1930
C CONH IS CONZ(I,J+1,K) SURT1940
C CONH = ZEMH(JG, I, K) SURT1950
C GET READY TO CALCULATE THE BOUNDARY TEMPERATURES OF THE GAP. SURT1960
C B = ZATIOH(JG, I, K)*CONL/CONH SURT1970
C C = T(I, J, K) + B*T(I, J + 1, K) SURT1980
C RGAP = 0.1713E-8*GAPZ(JG, I, K)/BX SURT1990
C ED = ZATIOH(JG)*CONH/GAPZ(JG, I, K) SURT2000
C ALLOW 10 ITERATIONS TO CALCULATE THE BOUNDARY TEMPERATURES. SURT2010
C DO 310 KK=1,10 SURT2020
C IF (KK.EQ.1) GO TO 290 SURT2030
C TH = ZBBTH(JG, I, K) SURT2040
C TB = ZBBTL(JG, I, K) SURT2050
C GO TO 300 SURT2060
290 TH = T(I, J, K) SURT2070
C TB = T(I, J + 1, K) SURT2080
300 GH = (GK + RGAP*(TH**2 + TB**2))*(TH + TB)*ED SURT2090
C X1 = GH*C SURT2100
C X2 = 1.0 + GH*(B + 1.0) SURT2110
C ZBBTL(JG, I, K) = (X1 + T(I, J + 1, K))/X2 SURT2120
C ZBBTH(JG, I, K) = (X1 + T(I, J, K))/X2 SURT2130
C ARE THE BOUNDARY TEMPERATURES CONVERGED ? -'YES,NO'- SURT2140
C IF (ABS(ZBBTH(JG,I,K)-TH).LE.1.0) GO TO 450 SURT2150
C HAVE THE 10 ITERATIONS BEEN PERFORMED ? -'YES,NO'- SURT2160
310 CONTINUE SURT2170
C NEGATIVE TEMPERATURES ARE PRESENT AND RESULT IN LOCAL SURT2180
C INSTABILITIES. TRY A SMALLER TIME STEP. SURT2190
C IERROR(1) = ISURT(1) SURT2200
C IERROR(2) = ISURT(3) SURT2210
C CALL ERROR2 SURT2220
C =====SURT2230
C CALCULATE THE BOUNDARY TEMPERATURES WHEN A COOLANT IS ADJACENT. SURT2240
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE ? -'NO,YES'- SURT2250
320 IF (GAPZ(J,I,K).LT.(-1.5E-10)) GO TO 380 SURT2260
C CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE HIGH SURT2270
C INDEX SIDE. SURT2280
C DXT = ZL(J) - ZP(J) SURT2290

```

C	RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE	SURT2300
C	PRECEEDING ITERATION AND PRESERVED IN THE ZEM-ZBBT ARRAYS.	SURT2310
C	CONL IS CONZ(I,J,K)	SURT2320
	CONL = ZEML(JG, I, K)	SURT2330
C	CONH IS CONZ(I,J+1,K)	SURT2340
	CONH = ZBBTL(JG, I, K)	SURT2350
C	EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.	SURT2360
	IF (JG.E3.NZG.OR.MATZG(JG,I,K).NE.200) GO TO 360	SURT2370
C	FIND THE HIGH AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT2380
	JG1 = JG + 1	SURT2390
	DO 340 JJ=JG1,NZG	SURT2400
	IF (GAPZ(JJ,I,K).LT.(-1.5E-10)) GO TO 330	SURT2410
	GO TO 340	SURT2420
C	EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT	SURT2430
C	THE HIGH AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT2440
330	IF (GAPZ(JJ,I,K).LT.(-2.5E-10)) GO TO 360	SURT2450
	JGH = JJ	SURT2460
	GO TO 350	SURT2470
340	CONTINUE	SURT2480
350	RAD = 0.1713E-8*(ZBBTH(JG, I, K)**4 - ZBBTL(JGH, I, K)**4)/(ZEMH(SURT2490
	JG, I, K) + ZEML(JGH, I, K) - 1.0)	SURT2500
C	PRESERVE RAD IN THE ZEML ARRAY.	SURT2510
	ZEML(JG, I, K) = RAD	SURT2520
	HX = CONH*(T(I, J, K) - T(I, J + 1, K) - RAD*CONL*DXT)/(T(I, J, K)	SURT2530
	- T(I, J + 1, K) + RAD*CONH)	SURT2540
	GO TO 370	SURT2550
360	HX = CONH	SURT2560
C	CALCULATE NEW BOUNDARY TEMPERATURES.	SURT2570
370	ZBBIH(JG, I, K) = (CONL*T(I, J + 1, K) + HX/DXT*T(I, J, K))/(CONL	SURT2580
	+ HX/DXT)	SURT2590
	ZBBIL(JG, I, K) = ZBBTH(JG, I, K)	SURT2600
	GO TO 450	SURT2610
C	ARE TWO COOLANTS ADJACENT ? -'NO,YES'-	SURT2620
380	IF (GAPZ(JG,I,K).GT.(-2.5E-10)) GO TO 390	SURT2630
C	DO NOT ATTEMPT TO CALCULATE BOUNDARY TEMPERATURES.	SURT2640
	GO TO 450	SURT2650
C	CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE LOW.	SURT2660
C	.INDX SIDE.	SURT2670
390	DXT = ZP(J + 1) - ZL(J)	SURT2680
C	RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE	SURT2690
C	PRECEEDING ITERATION AND PRESERVED IN THE ZEM-ZBBT ARRAYS.	SURT2700
C	CONL IS CONZ(I,J,K)	SURT2710
	CONL = ZBBTH(JG, I, K)	SURT2720
C	CONH IS CONZ(I,J+1,K)	SURT2730
	CONH = ZEMH(JG, I, K)	SURT2740
C	EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.	SURT2750


```

IF (JG.EQ.1) GO TO 430
IF (MATZG(JG,I,K).NE.200) GO TO 430
C FIND THE LOW AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
JG1 = JG - 1
DO 410 JJ=1,JG1
JG2 = JG - JJ
IF (GAPZ(JG2,I,K).LT.(-.5E-10)) GO TO 400
GO TO 410
C EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C THE LOW AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
400 IF (GAPZ(JG2,I,K).LT.(-2.5E-10)) GO TO 430
JGL = JG2
GO TO 420
410 CONTINUE
C RECOVER THE VALUE OF RAD PRESERVED IN THE ZEML ARRAY.
420 RAD = ZEML(JGL, I, K)
HX = CONL*(T(I, J + 1, K) - T(I, J, K) + CONH*RAD*DXT)/(T(I, J + 1, K) - T(I, J, K) - CONL*RAD)
GO TO 440
430 HX=CONL
C CALCULATE NEW BOUNDARY TEMPERATURES.
440 ZBRTL(JG, I, K) = (CONH*T(I, J, K) + HX/DXT*T(I, J + 1, K))/(CONH
+ HX/DXT)
ZBRTL(JG, I, K) = ZBRTL(JG, I, K)
C =====
C HAVE ALL POINTS ALONG THE CURRENT AXIAL GAPLINE BEEN
C CONSIDERED ? -'YES,NO'-
450 CONTINUE
C =====
C HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'-
460 CONTINUE
C =====
C HAVE ALL AXIAL BOUNDARY TEMPERATURES BEEN CALCULATED ?
C -'YES,NO'-
470 CONTINUE
C =====
C CALCULATE THE THETA BOUNDARY TEMPERATURES IN EACH THETA PLANE.
DO 660 K=1,KM
KG=KG1(K)
C =====
C EXCLUDE THE NON-GAPLINES.
IF (KG.LJ.0) GO TO 660
C =====
C CALCULATE THE THETA BOUNDARY TEMPERATURES FOR EACH AXIAL COLUMN
C IN THE PLANE.

```

```

SURT2760
SURT2770
SURT2780
SURT2790
SURT2800
SURT2810
SURT2820
SURT2830
SURT2840
SURT2850
SURT2860
SURT2870
SURT2880
SURT2890
SURT2900
SURT2910
SURT2920
SURT2930
SURT2940
SURT2950
SURT2960
SURT2970
SURT2980
SURT2990
SURT3000
SURT3010
SURT3020
SURT3030
SURT3040
SURT3050
SURT3060
SURT3070
SURT3080
SURT3090
SURT3100
SURT3110
SURT3120
SURT3130
SURT3140
SURT3150
SURT3160
SURT3170
SURT3180
SURT3190
SURT3200
SURT3210

```

```

C      DO 650 I=2,IM                                     SURT3220
C      =====SURT3230
C      CALCULATE THE THETA BOUNDARY TEMPERATURES IN THE COLUMN. SURT3240
C      =====SURT3250
C      DO 640 J=2,JM                                     SURT3260
C      IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS SURT3270
C      POINT ? -'NO,YES'-                               SURT3280
C      IF (GAPT(KG,I,J).EQ.0.0) GO TO 640              SURT3290
C      IS IT A COOLANT OR A GAP WHICH IS ADJACENT ? -,'COOLANT,GAP'- SURT3300
C      IF (GAPT(KG,I,J).LT.0.0) GO TO 510             SURT3310
C      =====SURT3320
C      CALCULATE THE BOUNDARY TEMPERATURES WHEN A GAP IS ADJACENT. SURT3330
C      RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE SURT3340
C      PRECEDING ITERATION AND PRESERVED IN THE TEM-TBBT ARRAYS. SURT3350
C      BX IS (TEMH(KG,I,J)+TEML(KG,I,J))-1.0)         SURT3360
C      BX = TBBTL(KG, I, J)                            SURT3370
C      GK IS THE GAS THERMAL CONDUCTIVITY             SURT3380
C      GK = TBBTH(KG, I, J)                            SURT3390
C      CONL IS CONT(I,J,K)                             SURT3400
C      CONL = TEML(KG, I, J)                           SURT3410
C      CONH IS CONT(I,J,K+1)                           SURT3420
C      CONH = TEMH(KG, I, J)                           SURT3430
C      GET READY TO CALCULATE THE BOUNDARY TEMPERATURES OF THE GAP. SURT3440
C      b = TATIOB(KG, I, J)*CONL/CONH                 SURT3450
C      c = T(I, J, K) + b*T(I, J, K + 1)             SURT3460
C      RGAP = 0.1713E-6*GAPT(KG, I, J)/BX           SURT3470
C      PUT RGAP IN THE CORRECT UNITS FOR CYLINDRICAL GEOMETRY. SURT3480
C      IF (ISHAPE.EQ.0) RGAP=RGAP*RP(I)              SURT3490
C      ED=1ATIOH(KG)*CONH/GAPT(KG,I,J)              SURT3500
C      ALLOW 10 ITERATIONS TO CALCULATE THE BOUNDARY TEMPERATURES. SURT3510
C      DO 500 KK=1,10                                  SURT3520
C      IF (KK.EQ.1) GO TO 480                          SURT3530
C      TH = TBBTH(KG, I, J)                            SURT3540
C      TB = TBBTL(KG, I, J)                            SURT3550
C      GO TO 490                                       SURT3560
480  TH = T(1, J, K)                                  SURT3570
      TB = T(1, J, K + 1)                              SURT3580
490  GH = (GK + RGAP*(TH**2 + TB**2))*(TH + TB))*ED   SURT3590
      X1 = GH*C                                         SURT3600
      X2 = 1.0 + GH*(b + 1.0)                          SURT3610
      TBBIL(KG, I, J) = (X1 + T(I, J, K + 1))/X2      SURT3620
      TBBIH(KG, I, J) = (X1 + T(I, J, K))/X2          SURT3630
C      ARE THE BOUNDARY TEMPERATURES CONVERGED ? -,'YES,NO'- SURT3640
C      IF (ABS(TBBTH(KG,I,J)-TH).LE.1.0) GO TO 640   SURT3650
C      HAVE THE 10 ITERATIONS BEEN PERFORMED ? -,'YES,NO'- SURT3660
500  CONTINUE                                         SURT3670

```

```

C      NEGATIVE TEMPERATURES ARE PRESENT AND RESULT IN LOCAL          SURT3680
C      INSTABILITIES. TRY A SMALLER TIME STEP.                          SURT3690
C      IERROR(1) = ISURT(1)                                             SURT3700
C      IERROR(2)=ISURT(4)                                             SURT3710
C      CALL ERROR2                                                    SURT3720
C      =====SURT3730
C      CALCULATE THE BOUNDARY TEMPERATURES WHEN A COOLANT IS ADJACENT. SURT3740
C      IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE ? -NO,YES'- SURT3750
510 IF (GAPT(KG,I,J).LT.(-1.5E-10)) GO TO 570                          SURT3760
C      CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE HIGH SURT3770
C      INDEX SIDE.                                                    SURT3780
C      DXT=TL(K)-TP(K)                                                 SURT3790
C      PUT DXT IN THE CORRECT UNITS FOR CYLINDRICAL GEOMETRY.         SURT3800
C      IF (ISHAPE.EQ.0) DXT=DXT*RP(I)                                  SURT3810
C      RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE SURT3820
C      PRECEDING ITERATION AND PRESERVED IN THE TEM-TBBT ARRAYS.     SURT3830
C      CONL IS CONT(I,J,K)                                             SURT3840
C      CONL = TEML(KG, I, J)                                           SURT3850
C      CONH IS CONT(I,J,K+1)                                           SURT3860
C      CONH = TBBTL(KG, I, J)                                          SURT3870
C      EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.    SURT3880
C      IF (KG.EQ.NTG.OK.MATTG(KG,I,J).NE.200) GO TO 550              SURT3890
C      FIND THE HIGH THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.    SURT3900
C      KG1=KG+1                                                        SURT3910
C      DO 530 KK=KG1,NTG                                               SURT3920
C      IF (GAPT(KK,I,J).LT.(-1.5E-10)) GO TO 520                      SURT3930
C      GO TO 530                                                       SURT3940
C      EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT SURT3950
C      THE HIGH THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.        SURT3960
520 IF (GAPT(KK,I,J).LT.(-2.5E-10)) GO TO 550                          SURT3970
C      KGH=KK                                                           SURT3980
C      GO TO 540                                                       SURT3990
530 CONTINUE                                                            SURT4000
540 RAD = 0.1713E-8*(TBBTH(KG, I, J)**4 - TBBTL(KGH, I, J)**4)/(TEMH SURT4010
C      * KG, I, J) + TEML(KGH, I, J) - 1.0)                          SURT4020
C      PRESERVE RAD IN THE TEML ARRAY.                                SURT4030
C      TEML(KG, I, J) = RAD                                           SURT4040
C      HX = CONH*(T(I, J, K) - T(I, J, K + 1) - RAD*CONL*DXT)/(T(I, J, K) SURT4050
C      - T(I, J, K + 1) + RAD*CONH)                                   SURT4060
C      GO TO 560                                                       SURT4070
550 HX = CONH                                                            SURT4080
C      CALCULATE NEW BOUNDARY TEMPERATURES.                          SURT4090
560 TBBTH(KG, I, J) = (CONL*T(I, J, K + 1) + HX/DXT*T(I, J, K))/(CONL SURT4100
C      + HX/DXT)                                                       SURT4110
C      TBBTL(KG, I, J) = TBBTH(KG, I, J)                              SURT4120
C      GO TO 640                                                       SURT4130

```

```

C ARE TWO COOLANTS ADJACENT ? -'NO,YES'- SURT4140
570 IF (GAP1(KG,I,J).GT.(-2.5E-10)) GO TO 580 SURT4150
C DO NOT ATTEMPT TO CALCULATE BOUNDARY TEMPERATURES. SURT4160
GO TO 640 SURT4170
C CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE LOW SURT4180
C INDEX SIDE. SURT4190
580 DXT=IP(K+1)-TL(K) SURT4200
C PUT DXT IN THE CORRECT UNITS FOR CYLINDRICAL GEOMETRY. SURT4210
IF (ISHAPE.EQ.U) DAT=DXT*RP(I) SURT4220
C RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE SURT4230
C PRECEDING ITERATION AND PRESERVED IN THE TEM-TBBT ARRAYS. SURT4240
CONL IS CONT(I,J,K) SURT4250
CONH=TBOTH(KG,I,J) SURT4260
C CONH IS CONT(I,J,K+1) SURT4270
CONH=TLWH(KG,I,J) SURT4280
C EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION. SURT4290
IF (KG.LQ.1) GO TO 620 SURT4300
IF (RATG(KG,I,J).NE.200) GO TO 620 SURT4310
C FIND THE LOW THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK. SURT4320
KG1=KG-1 SURT4330
DO DOU KK=1,KG1 SURT4340
KG2=KG-KK SURT4350
IF (GAP1(KG2,I,J).LT.(-.5E-10)) GO TO 590 SURT4360
GO TO 600 SURT4370
C EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT SURT4380
C THE LOW THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK. SURT4390
590 IF (GAP1(KG2,I,J).LT.(-2.5E-10)) GO TO 620 SURT4400
KGL=KG2 SURT4410
GO TO 610 SURT4420
600 CONTINUE SURT4430
C RECOVER THE VALUE OF RAD PRESERVED IN THE TEMPL ARRAY. SURT4440
610 RAD = TEMPL(KGL, I, J) SURT4450
HX = CONL*(T(I, J, K + 1) - T(I, J, K) + CONH*RAD*DXT)/(T(I, J, K SURT4460
+ 1) - T(I, J, K) - CONL*RAD) SURT4470
GO TO 630 SURT4480
620 HX=CONL SURT4490
C CALCULATE NEW BOUNDARY TEMPERATURES. SURT4500
630 TBBIL(KG, I, J) = (CONH*T(I, J, K) + HX/DXT*T(I, J, K + 1))/(CONH SURT4510
+ HX/DXT) SURT4520
TBBIH(KG, I, J) = TBBTL(KG, I, J) SURT4530
C ===== SURT4540
C HAVE ALL POINTS ALONG THE CURRENT THETA GAPLINE BEEN SURT4550
C CONSIDERED ? -'YES,NO'- SURT4560
640 CONTINUE SURT4570
C ===== SURT4580
C HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'- SURT4590
650 CONTINUE SURT4600
C ===== SURT4610
C HAVE ALL THETA BOUNDARY TEMPERATURES BEEN CALCULATED ? -'YES,NO'- SURT4620
660 CONTINUE SURT4630
C ===== SURT4640
RETURN SURT4650
END SURT4660

```

```

SUBROUTINE PRINT
INCLUDE COMDIM
=====PRIN 10
=====PRIN 20
=====PRIN 30
=====PRIN 40
INTEGER TT PRIN 50
LOGICAL WRES PRIN 60
DIMENSION DTO (NQ),DTI (NQ),WAY ( 6), PRIN 70
1 AT ( IQ,JQ,KQ), CT ( IQ,JQ,KQ), PRIN 80
3 DDT ( IQ,JQ,KQ) PRIN 90
DATA (WAY(I),I=1,6) /6H+1,J,K, 6H,J+1,K, 6H,J,K+1, 3H(I), 3H(J), PRIN 100
- 3H(K)/ PRIN 110
EQUIVALENCE (TT ,AT ), PRIN 120
1 (TT ,CT ),(TT ,DDT ),(ITO ,DTO ), PRIN 130
2 (ITI ,DTI ) PRIN 140
C= =====PRIN 150
C= =====PRIN 160
IRET=0 PRIN 170
WRES=FALSE. PRIN 180
X1=CURTI*00.0 PRIN 190
X2=CURTI*3000.0 PRIN 200
C SELECT THE TEMPERATURE DISTRIBUTION PRINT FORMAT PRIN 210
IF(LASTIP.EQ.NITER) GO TO 190 PRIN 220
IF (SW(9)) GO TO 190 PRIN 230
C= =====PRIN 240
C PREPARE AND PRINT COOLANT AND NO2E TEMPERATURE DISTRIBUTION IN PRIN 250
C INTEGRAL DEGREES FAHRENHEIT PRIN 260
GO TO 167 PRIN 270
ENTRY TEMPS PRIN 280
IRET=1 PRIN 290
167 WRITE(6,110) (ZA(I),I=1,12) PRIN 300
110 FORMAT (1H1,30X,12A6,///) PRIN 310
IF (IRET.EQ.2) GO TO 169 PRIN 320
WRITE (6,111) PRIN 330
111 FORMAT (49X,2AHCOOLANT TEMPERATURES (F),/,3X, PRIN 340
114HCOOLANT NUMBER,4X,5HINLET,5X,6HOUTLET,4X,12HFLOW (LB/HR),8X,14H PRIN 350
2COOLANT NUMBER,4X,5HINLET,5X,6HOUTLET,4X,12HFLOW (LB/HR)) PRIN 360
C PREPARE ALL COOLANT DATA FOR PRINTING PRIN 370
DO 150 N=1,MAXFLO,2 PRIN 380
IF (NFLO(N).EQ.0) GO TO 150 PRIN 390
N1=NFLO(N) PRIN 400
N2=NFLO(N+1) PRIN 410
ITI(N1)=TI(N1)-459.5 PRIN 420
ITO(N1)=TO(N1)-459.5 PRIN 430
IF (N1)=FLOW(N1) PRIN 440
IF (N2.EQ.0 .OR. N.EQ.15) GO TO 120 PRIN 450

```

```

      ITI(N2)=TI(N2)-459.5                                PRIN 460
      ITO(N2)=TO(N2)-459.5                                PRIN 470
      IF(N2)=FLOW(N2)                                     PRIN 480
      WRITE (6,140) N1,ITI(N1),ITO(N1),IF(N1),N2,ITI(N2),ITO(N2),IF(N2) PRIN 490
140  FORMAT(BX,I2,I16,I11,I14,15X,I2,I16,I11,I14)         PRIN 500
      GO TO 130                                           PRIN 510
120  WRITE (6,140) N1,ITI(N1),ITO(N1),IF(N1)             PRIN 520
      GO TO 130                                           PRIN 530
130  CONTINUE                                           PRIN 540
150  IF(1RET.EQ.0) GO TO 153                             PRIN 550
109  WRITE(6,155) NITER                                  PRIN 560
155  FORMAT(1H0,3AX,I4,51H ITERATIONS PERFORMED WITHOUT REACHING STEADYPRIN 570
      . STATE,/)
      IF(1RET.EQ.2) RETURN                                PRIN 580
      GO TO 157                                           PRIN 600
153  WRITE(6,160) CURTI,X1,X2,NITER                     PRIN 610
160  FORMAT (1H0,21H THE CURRENT TIME IS,F10.4,11H HOURS = ,F10.4,13PRIN 620
      1H MINUTES = ,F15.5,8H SECONDS,4X,I4,31H ITERATIONS HAVE BEEN PERPRIN 630
      2FORMAT,D,/)                                       PRIN 640
C=====PRIN 650
C CONVERT TEMPERATURES TO INTEGRAL DEGREES FAHRENHEIT PRIN 660
157  DO 170 I=1,IMAX PRIN 670
      DO 170 J=1,JMAX PRIN 680
      DO 170 K=1,KMAX PRIN 690
      TT(I,J,K)=T(I,J,K)-459.5 PRIN 700
170  CONTINUE PRIN 710
C SET THE MESH RIBS IN THE PRINTOUT EQUAL TO ZERO PRIN 720
      DO 516 L = 1,JMAX PRIN 730
      TT(1,L,1) = 0 PRIN 740
      TT(IMAX,L,1) = 0 PRIN 750
      TT(1,L,KMAX) = 0 PRIN 760
516  TT(IMAX,L,KMAX) = 0 PRIN 770
      DO 525 L = 1,IMAX PRIN 780
      TT(L,1,1) = 0 PRIN 790
      TT(L,JMAX,1) = 0 PRIN 800
      TT(L,1,KMAX) = 0 PRIN 810
525  TT(L,JMAX,KMAX) = 0 PRIN 820
      DO 553 L = 1,KMAX PRIN 830
      TT(1,1,L) = 0 PRIN 840
      TT(IMAX,1,L) = 0 PRIN 850
      TT(1,JMAX,L) = 0 PRIN 860
553  TT(IMAX,JMAX,L) = 0 PRIN 870
C=====PRIN 880
C=====PRIN 890
C PRINT THE TEMPERATURE (INTEGRAL DEGREES F) PRIN 900
      WRITE (6,180) PRIN 910

```

```

160 FORMAT (1H0,47X,20HTEMPERATURES      (F))          PRIN 920
    CALL ARRAY (TI,DUM,1)                               PRIN 930
    IF (IRET.EQ.0) GO TO 159                            PRIN 940
    IRET=2                                               PRIN 950
    GO TO 167                                            PRIN 960
159 IF (ILKRROR(1).GT.0) WRITE(6,185) IERROR(1),NITER PRIN 970
185 FORMAT(1H0,5H*****,14,52H ERROR MESSAGES PRECEED THE RESULTS OF ITPRIN 980
    .ERATION NO.,I4,5H*****)
    LASTIP=NITER                                        PRIN 990
    IF (.NOT. DP) GO TO 280                              PRIN1010
C= =====PRIN1020
C   PREPARE AND PRINT THE COOLANT AND NODE TEMPERATURE DISTRIBUTION PRIN1030
C   IN DECIMAL DEGREES FAHRENHEIT                      PRIN1040
190 WRITE (6,110) (ZA(I),I=1,12)                       PRIN1050
    WRITE (6,200)                                       PRIN1060
200 FORMAT      (49X,28HCOOLANT TEMPERATURES      (F),/,1X,PRIN1070
114HCOOLANT NUMBER,5X,5HINLET,11X,6HOUTLET,7X,12HFLOW (LB/HR),4X,14PRIN1080
2HCOOLANT NUMBER,5X,5HINLET,11X,6HOUTLET,7X,12HFLOW (LB/HR)) PRIN1090
    DO 230 N=1,MAXFLO,2                                  PRIN1100
    IF (NFLO(N).EQ.0) GO TO 250                          PRIN1110
    N1=NFLO(N)                                          PRIN1120
    N2=NFLO(N+1)                                        PRIN1130
    DTI(N1)=TI(N1)-460.0                                PRIN1140
    IF (DTI(N1) .EQ. 0.0) DTI(N1)=0.0                  PRIN1150
    DTO(N1)=TO(N1)-460.0                                PRIN1160
    IF (DTO(N1) .EQ. 0.0) DTO(N1)=0.0                  PRIN1170
    IF (N2.EQ.0 .OR. N.EQ.15) GO TO 220                 PRIN1180
    DTI(N2)=TI(N2)-460.0                                PRIN1190
    IF (DTI(N2) .EQ. 0.0) DTI(N2)=0.0                  PRIN1200
    DTO(N2)=TO(N2)-460.0                                PRIN1210
    IF (DTO(N2) .EQ. 0.0) DTO(N2)=0.0                  PRIN1220
    WRITE (6,240) N1,DTI(N1),DTO(N1),FLOW(N1),N2,DTI(N2),DTO(N2),FLOW(PRIN1230
    N2)                                                  PRIN1240
240 FORMAT (7X,I2,7X,1PE13.7,3X,E13.7,3X,E13.7,I12,7X,E13.7,3X,E13.7,3PRIN1250
    1X,E13.7)                                           PRIN1260
    GO TO 230                                            PRIN1270
220 WRITE (6,240) N1,DTI(N1),DTO(N1),FLOW(N1)         PRIN1280
    GO TO 250                                            PRIN1290
230 CONTINUE                                           PRIN1300
250 WRITE (6,160) CURTI,X1,X2,NITER                   PRIN1310
C= =====PRIN1320
C   CONVERT THE TEMPERATURES TO DEGREES FAHRENHEIT    PRIN1330
    DO 260 I=1,I,MAX                                    PRIN1340
    DO 260 J=1,J,MAX                                    PRIN1350
    DO 260 K=1,K,MAX                                    PRIN1360
    DDT(I,J,K)=T(I,J,K)-460.0                          PRIN1370

```

```

IF (DDT(I,J,K) .EQ. 0.0) DDT(I,J,K)=0.0 PRIN13A0
C 260 CONTINUE PRIN1390
SET THE MESH RIBS IN THE PRINTOUT EQUAL TO ZERO PRIN1400
DO 501 L = 1,JMAX PRIN1410
DDT(1,L,1)=0.0 PRIN1420
DDT(IMAX,L,1)=0.0 PRIN1430
DDT(1,L,KMAX)=0.0 PRIN1440
501 DDT(IMAX,L,KMAX)=0.0 PRIN1450
DO 526 L=1,IMAX PRIN1460
DDT(L,1,1)=0.0 PRIN1470
DDT(L,JMAX,1)=0.0 PRIN1480
DDT(L,1,KMAX)=0.0 PRIN1490
526 DDT(L,JMAX,KMAX)=0.0 PRIN1500
DO 551 L=1,KMAX PRIN1510
DDT(1,1,L)=0.0 PRIN1520
DDT(IMAX,1,L)=0.0 PRIN1530
DDT(1,JMAX,L)=0.0 PRIN1540
551 DDT(IMAX,JMAX,L)=0.0 PRIN1550
C=====PRIN1560
C= PRIN1570
C PRINT THE TEMPERATURES (DECIMAL DEGREES F) PRIN15A0
WRITE (6,270) PRIN1590
270 FORMAT (1H0,47X,28HDECIMAL TEMPERATURES (F)) PRIN1600
CALL ARRAY (IDUM,DDT,3) PRIN1610
IF (IERROR(1).GT.0) WRITE(6,185)IERROR(1),NITER PRIN1620
C=====PRIN1630
C CALCULATE AND PRINT THE RESIDUALS FOR THE LAST ITERATION OF A PRIN1640
C STEADY STATE PROBLEM. PRIN1650
IF (DP.AND.SN(10)) GO TO 273 PRIN1660
GO TO 280 PRIN1670
273 NRES=.TRUE. PRIN16A0
GO TO 355 PRIN1690
275 CALL RESID PRIN1700
NRES=.FALSE. PRIN1710
C=====PRIN1720
C DO WHAT THE USER WISHES TO DO PRIN1730
280 CALL CUSTOM PRIN1740
C=====PRIN1750
C SHOULD THE SURFACE TEMPERATURES BE PRINTED? -1NO,YES1- PRIN1760
IF (.NOT.SN(4)) GO TO 340 PRIN1770
C=====PRIN17A0
C PRINT THE RADIAL-X SURFACE TEMPERATURES (LOW AND HIGH) PRIN1790
WRITE (6,350) (ZA(I),I=1,12),CURT1,X1,X2,NITER PRIN1800
WRITE (6,290) PNAME(1),PNAME(2),WAY(6),PNAME(3),WAY(5) PRIN1810
290 FORMAT (19X,32HSURFACE TEMPERATURES AT THE A6, PRIN1820
'348H GAP OR COOLANT BOUNDARY GRIDPLANES (F),//,43X,4HTHE ,PRIN1830

```



```

X A6, A3, 32H DIRECTION POINTS ARE HORIZONTAL/43X, 4HTHE , A6, A3, 30H DIRPRIN1840
SECTION POINTS ARE VERTICAL) PRIN1850
C CONVERT THE RADIAL-X SURFACE TEMPERATURES TO INTEGRAL DEGREES ° PRIN1860
C FAHRENHEIT, AND INSURE THAT A POINT IN THE GRIDPLANE WHICH DOES PRIN1870
C NOT BELONG TO A GAP OR COOLANT SURFACE APPEARS AS A BLANK IN PRIN1880
C THE PRINT OUT PRIN1890
CALL GPRINT (IGR, IM, PNAME(1), JMAX, KMAX, IGQ, JQ, KQ, RBBTH, RBBTL, GAPR) PRIN1900
C= =====PRIN1910
C PRINT THE AXIAL-Z SURFACE TEMPERATURES (LOW AND HIGH) PRIN1920
WRITE (6, 360) (ZA(I), I=1, 12), CURTI, X1, X2, NITER PRIN1930
WRITE (6, 290) PNAME(2), PNAME(1), WAY(4), PNAME(3), WAY(6) PRIN1940
C CONVERT THE AXIAL-Z SURFACE TEMPERATURES TO INTEGRAL DEGREES PRIN1950
C FAHRENHEIT, AND INSURE THAT A POINT IN THE GRIDPLANE WHICH DOES PRIN1960
C NOT BELONG TO A GAP OR COOLANT SURFACE APPEARS AS A BLANK IN PRIN1970
C THE PRINT OUT PRIN1980
CALL GPRINT (JGZ, JM, PNAME(2), IMAX, KMAX, JGQ, IQ, KQ, ZBBTH, ZBBTL, GAPZ) PRIN1990
C= =====PRIN2000
C PRINT THE THETA-Y SURFACE TEMPERATURES (LOW AND HIGH) PRIN2010
WRITE (6, 360) (ZA(I), I=1, 12), CURTI, X1, X2, NITER PRIN2020
WRITE (6, 290) PNAME(3), PNAME(1), WAY(4), PNAME(2), WAY(5) PRIN2030
C CONVERT THE THETA-Y SURFACE TEMPERATURES TO INTEGRAL DEGREES PRIN2040
C FAHRENHEIT, AND INSURE THAT A POINT IN THE GRIDPLANE WHICH DOES PRIN2050
C NOT BELONG TO A GAP OR COOLANT SURFACE APPEARS AS A BLANK IN PRIN2060
C THE PRINT OUT PRIN2070
CALL GPRINT (KGT, KM, PNAME(3), IMAX, JMAX, KGQ, IQ, JQ, TBBTH, TBBTL, GAPY) PRIN2080
C= =====PRIN2090
C SHOULD THE HEAT RATES AND HEAT FLUXES BE PRINTED? -'NO, YES'- PRIN2100
340 IF (.NOT.SW(5)) GO TO 500 PRIN2110
C= =====PRIN2120
C CALCULATE THE RADIAL-X HEAT RATES PRIN2130
DO 350 K=1, KMAX PRIN2140
DO 350 J=1, JMAX PRIN2150
DO 350 I=1, IMAX PRIN2160
AT(I, J, K)=RR(I, J, K)*KR(I, J, K)*(T(I, J, K)-T(I+1, J, K)) PRIN2170
350 CONTINUE PRIN2180
C PRINT THE RADIAL-X HEAT RATES PRIN2190
355 WRITE(6, 360) (ZA(I), I=1, 12), CURTI, X1, X2, NITER PRIN2200
360 FORMAT (1H1, 30X, 12A6, //, 21H THE CURRENT TIME IS, F10.4, 11H HOURS PRIN2210
1= , F10.4, 13H MINUTES = , F13.5, 8H SECONDS, 4X, I4, 31H ITERATIONS HPRIN2220
2AVE BEEN PERFORMED, ///) PRIN2230
IF(NRES) GO TO 275 PRIN2240
WRITE (6, 370) PNAME(1), WAY(1) PRIN2250
370 FORMAT ( 19X, 15HHEAT RATE IN , A6, 46H DIRECTION BETWEEN P0I PRIN2260
1, TS (I, J, K) AND (I, A6, 14H) (BTU/HR)) PRIN2270
CALL ARRAY (IDUM, AT, 2) PRIN2280
C= =====PRIN2290

```

```

C      CALCULATE THE RADIAL-X HEAT FLUXES                                PRIN2300
      NAME=ISHAPE+1                                                    PRIN2310
      DO 410 K=1,KMAX                                                  PRIN2320
      DO 410 J=1,JMAX                                                  PRIN2330
      DO 410 I=1,IM                                                    PRIN2340
      GO TO (380,390),NAME                                             PRIN2350
380    AT(I,J,K)=AT(I,J,K)/(RR(I,J,K)*RL(I)*ALOG(RP(I+1)/RP(I)))    PRIN2360
      GO TO 410                                                        PRIN2370
390    AT(I,J,K)=AT(I,J,K)/(RR(I,J,K)*DELR(I))                      PRIN2380
410    CONTINUE                                                       PRIN2390
C      PRINT THE RADIAL-X HEAT FLUXES                                  PRIN2400
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER                 PRIN2410
      WRITE (6,420) PNAME(1),WAY(4),WAY(1)                            PRIN2420
420    FORMAT ( 19X,16HHEAT FLUX IN ,A6,48H DIRECTION BASED ON PRIN2430
1THE AREA OF GRIDLINE ,A3,/,28X,33HBETWEEN POINTS (I,J,K) ANDPRIN2440
2 (1,A6,20H) (BTU/HR-FT**3))                                         PRIN2450
      CALL ARRAY (IDUM,AT,2)                                           PRIN2460
C=  =====PRIN2470
C      CALCULATE THE AXIAL-Z HEAT RATE                                PRIN2480
      DO 430 K=1,KMAX                                                  PRIN2490
      DO 430 J=1,JMAX                                                  PRIN2500
      DO 430 I=1,IMAX                                                  PRIN2510
      AT(I,J,K)=KZ(I,J,K)*KZ(I,J,K)*(T(I,J,K)-T(I,J+1,K))          PRIN2520
430    CONTINUE                                                       PRIN2530
C      PRINT THE AXIAL-Z HEAT RATE                                    PRIN2540
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER                 PRIN2550
      WRITE (6,370) PNAME(2),WAY(2)                                    PRIN2560
      CALL ARRAY (IDUM,AT,2)                                           PRIN2570
C=  =====PRIN2580
C      CALCULATE THE AXIAL-Z HEAT FLUXES                              PRIN2590
      DO 440 J=1,JMAX                                                  PRIN2600
      DO 440 I=1,IMAX                                                  PRIN2610
      DO 440 K=1,KMAX                                                  PRIN2620
      GO TO (450,460),NAME                                             PRIN2630
450    AT(I,J,K)=AT(I,J,K)/(RZ(I,J,K)*(7P(J+1)-2P(J)))              PRIN2640
      GO TO 440                                                        PRIN2650
460    AT(I,J,K)=AT(I,J,K)/(RZ(I,J,K)*DELZ(J))                      PRIN2660
480    CONTINUE                                                       PRIN2670
C      PRINT THE AXIAL-Z HEAT FLUXES                                  PRIN2680
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER                 PRIN2690
      WRITE (6,420) PNAME(2),WAY(5),WAY(2)                            PRIN2700
      CALL ARRAY (IDUM,AT,2)                                           PRIN2710
C=  =====PRIN2720
C=  =====PRIN2730
C      CALCULATE THE THETA-Y HEAT RATE                                PRIN2740
      DO 1430 K=1,KMAX                                                 PRIN2750

```

```

DO 1430 J=1,JMAX                                PRIN2760
DO 1430 I=1,IMAX                                PRIN2770
AT(I,J,K)=RT(I,J,K)*KT(I,J,K)*(T(I,J,K)-T(I,J,K+1)) PRIN27A0
1430 CONTINUE                                    PRIN2790
C PRINT THE THETA-Y HEAT RATE                    PRIN2800
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER PRIN2810
WRITE (6,370) PNAME(3),WAY(3)                  PRIN2820
CALL ARRAY (IDUM,AT,2)                           PRIN2830
C= =====PRIN2840
C CALCULATE THE THETA-Y HEAT FLUXES              PRIN2850
DO 1460 K=1,KM                                  PRIN2860
DO 1430 J=1,JMAX                                PRIN2870
DO 1460 I=1,IMAX                                PRIN2880
GO TO (1450,1460),NAME                          PRIN2890
1450 AT(I,J,K)=AT(I,J,K)/(RT(I,J,K)*(RP(I)*(TP(K+1)-TP(K)))) PRIN2900
GO TO 1480                                        PRIN2910
1460 AT(I,J,K)=AT(I,J,K)/(RT(I,J,K)*DELTA(K)) PRIN2920
1480 CONTINUE                                    PRIN2930
C PRINT THE THETA-Y HEAT FLUXES                  PRIN2940
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER PRIN2950
WRITE (6,420) PNAME(3),WAY(6),WAY(3)           PRIN2960
CALL ARRAY (IDUM,AT,2)                           PRIN2970
C= =====PRIN2980
C SHOULD THE EFFECTIVE THERMAL CONDUCTIVITIES AND THERMAL PRIN2990
C CONDUCTANCES BE PRINTED? -NO,YES'-           PRIN3000
500 IF (.NOT.SW(7)) GO TO 610                    PRIN3010
C= =====PRIN3020
DO 510 K=1,KMAX                                  PRIN3030
DO 510 I=1,IMAX                                  PRIN3040
KR(I,1,K)=0.0                                    PRIN3050
150 KR(I,JMAX,K)=0.0                              PRIN3060
DO 515 I=1,IMAX                                  PRIN3070
DO 515 J=1,JMAX                                  PRIN3080
KR(I,J,1)=0.0                                    PRIN3090
155 KR(I,J,KMAX)=0.0                              PRIN3100
DO 520 K=1,KMAX                                  PRIN3110
DO 520 J=1,JMAX                                  PRIN3120
520 KR(IMAX,J,K)=0.0                              PRIN3130
C PRINT THE EFFECTIVE RADIAL-X THERMAL CONDUCTIVITIES PRIN3140
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER PRIN3150
WRITE (6,530) PNAME(1),WAY(1)                   PRIN3160
530 FORMAT ( 19X,11HEFFECTIVE ,A6,49H CONDUCTIVITY BETWEEN POINTS PRIN3170
1TS (1,J,K) AND (I,A6,19H) (BTU/HR-FT-F)) PRIN3180
CALL ARRAY (IDUM,KR,2)                           PRIN3190
C= =====PRIN3200
DO 540 K=1,KMAX                                  PRIN3210

```

```

DO 540 I=1,IMAX
540 KZ(1,JMAX,K)=0.0
DO 545 I=1,IMAX
DO 545 J=1,JMAX
KZ(1,J,1)=0.0
545 KZ(1,J,KMAX)=0.0
DO 550 K=1,KMAX
DO 550 J=1,JMAX
KZ(1,J,K)=0.0
550 KZ(IMAX,J,K)=0.0
C PRINT THE EFFECTIVE AXIAL-Z THERMAL CONDUCTIVITIES
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
WRITE (6,330) PNAME(2),WAY(2)
CALL ARRAY (IDUM,KZ,2)
C=
=====
DO 1540 I=1,IMAX
DO 1540 J=1,JMAX
1540 KT(1,J,KMAX)=0.0
DO 1545 K=1,KMAX
DO 1545 J=1,JMAX
KT(1,J,K)=0.0
1545 KT(IMAX,J,K)=0.0
DO 1550 K=1,KMAX
DO 1550 I=1,IMAX
KT(1,I,K)=0.0
1550 KT(1,JMAX,K)=0.0
C PRINT THE EFFECTIVE THETA-Y THERMAL CONDUCTIVITIES
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
WRITE (6,330) PNAME(3),WAY(3)
CALL ARRAY (IDUM,KT,2)
C=
=====
DO 570 I=1,IMAX
DO 570 J=1,JMAX
DO 570 K=1,KMAX
CT(1,J,K)=KR(I,J,K)*KR(I,J,K)
570 CONTINUE
C PRINT THE RADIAL-X THERMAL CONDUCTANCES
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
WRITE (6,380) PNAME(1),WAY(1)
580 FORMAT ( 19X,A6,57H THERMAL CONDUCTANCE BETWEEN POINTS (I,
1J,K) AND (I,A6,16H) (BTU/HR-F))
CALL ARRAY (IDUM,CT,2)
C=
=====
DO 590 I=1,IMAX
DO 590 J=1,JMAX
DO 590 K=1,KMAX

```

```

PRIN3220
PRIN3230
PRIN3240
PRIN3250
PRIN3260
PRIN3270
PRIN3280
PRIN3290
PRIN3300
PRIN3310
PRIN3320
PRIN3330
PRIN3340
PRIN3350
PRIN3360
PRIN3370
PRIN3380
PRIN3390
PRIN3400
PRIN3410
PRIN3420
PRIN3430
PRIN3440
PRIN3450
PRIN3460
PRIN3470
PRIN3480
PRIN3490
PRIN3500
PRIN3510
PRIN3520
PRIN3530
PRIN3540
PRIN3550
PRIN3560
PRIN3570
PRIN3580
PRIN3590
PRIN3600
PRIN3610
PRIN3620
PRIN3630
PRIN3640
PRIN3650
PRIN3660
PRIN3670

```

```

CT(I,J,K)=RZ(I,J,K)*KZ(I,J,K) PRIN36A0
590 CONTINUE PRIN3690
C PRINT THE AXIAL-Z THERMAL CONDUCTANCES PRIN3700
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER PRIN3710
WRITE (6,580) PNAME(2),WAY(2) PRIN3720
CALL ARRAY (IDUM,CT,2) PRIN3730
C=====PRIN3740
DO 1590 I=1,IMAX PRIN3750
DO 1590 J=1,JMAX PRIN3760
DO 1590 K=1,KMAX PRIN3770
CT(I,J,K)=RT(I,J,K)*KT(I,J,K) PRIN37A0
1590 CONTINUE PRIN3790
C PRINT THE THETA-Y THERMAL CONDUCTANCES PRIN3800
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER PRIN3810
WRITE (6,580) PNAME(3),WAY(3) PRIN3820
CALL ARRAY (IDUM,CT,2) PRIN3830
C=====PRIN3840
C SHOULD THE OUTPUT BE PUT ON A TAPE? -'NO,YES,- PRIN3850
610 IF (.NOT.S*(3)) GO TO 620 PRIN3860
C=====PRIN3870
C PUT THE CURRENT TEMPERATURE DISTRIBUTION ON TAPE PRIN38A0
WRITE (OUTAP) IMAX,JMAX,KMAX,MAXFLO,CURTI,NITER, PRIN3890
.(FLO(N),TI(N),TO(N),N=1,MAXFLO), PRIN3900
.(((T(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX) PRIN3910
C RESTORE THE SECOND TEMPERATURE ARRAY PRIN3920
C=====PRIN3930
620 IF (.NOT.OP) GO TO 640 PRIN3940
WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER PRIN3950
WRITE (6,1000) PRIN3960
1000 FORMAT (52X,15HHEAT BALANCE,///,29X,15HHEAT GENERATED,25X, PRIN3970
X28HEAT LOST/GAINED BY COOLANTS, //, 25X,12HBLOCK NUMBER,9X, PRIN39A0
X6HBTU/HR,13X,14HCOOLANT NUMBER,8X,6HBTU/HR,/) PRIN3990
SUM=0.0 PRIN4000
SUMC=0.0 PRIN4010
DO 1200 L=1,LMAX PRIN4020
ILS=IL(L) PRIN4030
IHS=IH(L) PRIN4040
JLS=JL(L) PRIN4050
JHS=JH(L) PRIN4060
KLS=KL(L) PRIN4070
KHS=KH(L) PRIN4080
SUM=0.0 PRIN4090
C TEST WHETHER A BLOCK IS A COOLANT OR NOT PRIN4100
IF (MB(L).GT.0) GO TO 1100 PRIN4110
N=IABS(MB(L)) PRIN4120
I=IABS(IPATH(N)) PRIN4130

```

	GO TO (1025,1050,1010), II	PRIN4140
C	AXIAL-Z COOLANT FLOW	PRIN4150
1050	DO 1060 J=JLS,JHS	PRIN4160
	DO 1065 K=KLS,KHS	PRIN4170
	IF (ILS .LE. 2) GO TO 1055	PRIN4180
	I=ILS-1	PRIN4190
	SUM =SUM + RR(I,J,K) * KR(I,J,K) * (T(I,J,K)-T(ILS,J,K))	PRIN4200
1055	IF (IHS .GE. IM) GO TO 1065	PRIN4210
	SUM = SUM - RR(IHS,J,K) * KR(IHS,J,K) * (T(IHS,J,K) - T(IHS+1,J,K))	PRIN4220
1065	CONTINUE	PRIN4230
	DO 1068 I=ILS,IHS	PRIN4240
	IF (KLS .LE. 2) GO TO 1069	PRIN4250
	K=KLS-1	PRIN4260
	SUM =SUM + RT(I,J,K) * KT(I,J,K) * (T(I,J,K) - T(I,J,KLS))	PRIN4270
1069	IF (KHS .GE. KM) GO TO 1068	PRIN4280
	SUM = SUM - RT(I,J,KHS) * KT(I,J,KHS) * (T(I,J,KHS) - T(I,J,KHS+1))	PRIN4290
1068	CONTINUE	PRIN4300
1060	CONTINUE	PRIN4310
	GO TO 1075	PRIN4320
C	RADIAL-X COOLANT FLOW	PRIN4330
1025	DO 1040 I=ILS,IHS	PRIN4340
	DO 1030 K=KLS,KHS	PRIN4350
	IF (JLS .LE. 2) GO TO 1028	PRIN4360
	J=JLS-1	PRIN4370
	SUM = SUM + RZ(I,J,K) * KZ(I,J,K) * (T(I,J,K) - T(I,JLS,K))	PRIN4380
1028	IF (JHS .GE. JM) GO TO 1030	PRIN4390
	SUM = SUM - RZ(I,JHS,K) * KZ(I,JHS,K) * (T(I,JHS,K)-T(I,JHS+1,K))	PRIN4400
1030	CONTINUE	PRIN4410
	DO 1035 J=JLS,JHS	PRIN4420
	IF (KLS .LE. 2) GO TO 1032	PRIN4430
	K=KLS-1	PRIN4440
	SUM = SUM + RT(I,J,K) * KT(I,J,K) * (T(I,J,K) - T(I,J,KLS))	PRIN4450
1032	IF (KHS .GE. KM) GO TO 1035	PRIN4460
	SUM = SUM - RT(I,J,KHS)*KT(I,J,KHS)*(T(I,J,KHS)-T(K,J,KHS+1))	PRIN4470
1035	CONTINUE	PRIN4480
1040	CONTINUE	PRIN4490
	GO TO 1075	PRIN4500
C	THETA-Y COOLANT FLOW	PRIN4510
1010	DO 1020 K=KLS,KHS	PRIN4520
	DO 1015 J=JLS,JHS	PRIN4530
	IF (ILS .LE. 2) GO TO 1012	PRIN4540
	I=ILS-1	PRIN4550
	SUM=SUM + RR(I,J,K) * KR(I,J,K) * (T(I,J,K)-T(ILS,J,K))	PRIN4560
1012	IF (IHS.GE. IM) GO TO 1015	PRIN4570
	SUM = SUM-RR(IHS,J,K) * KR(IHS,J,K) * (T(IHS,J,K)-T(IHS+1,J,K))	PRIN4580
1015	CONTINUE	PRIN4590

```

DO 1024 I=JLS,IHS
IF (JLS .LE. 2) GO TO 1022
J=JLS-1
SUM=SUM + RZ(I,J,K) * KZ(I,J,K) * (T(I,J,K)-T(I,JLS,K))
1022 IF (JHS .GE. JM) GO TO 1024
SUM = SUM -RZ(I,JHS,K) * KZ(I,JHS,K) * (T(I,JHS,K) - T(I,JHS+1,K))
1024 CONTINUE
1020 CONTINUE
1075 SUMC=SUMC + SUM
WRITE (6,1080) N,SUM
1080 FORMAT (69X,I4,9X,1PE14.7)
GO TO 1200
1100 DO 1100 I=JLS,IHS
DO 1100 J=JLS,JHS
DO 1100 K=KLS,KHS
SUM = SUM + w(I,J,K)
1100 CONTINUE
SUMM = SUMM + SUM
IF (SUM .EQ. 0.0) GO TO 1200
WRITE (6,1170) MB(L),SUM
1170 FORMAT (28X,I4,9X,1PE14.7)
1200 CONTINUE
WRITE (6,1205) SUMM,SUMC
1205 FORMAT (1H0,39X,16H-----,25X,16H-----,/,
X41X,1PE14.7,27X,1PE14.7)
C =====PRIN4850
640 DO 630 J=1,JMAX
DO 630 K=1,KMAX
DO 630 I=1,IMAX
AT(I,J,K)=T(I,J,K)
630 CONTINUE
RETURN
PRIN4910
C= =====PRIN4920
C *****PRIN4930
C *****PRIN4940
SUBROUTINE GPRINT(NGAP,NGRID,PLAHL,ICROSS,IDOWN,N1,N2,N3,BBTH,
XBBTL,GAP)
PRIN4950
C= =====PRIN4970
DIMENSION NGAP ( 1), BBTH(N1,N2,N3), BBTL(N1,N2,N3),
PRIN4980
X GAP(N1,N2,N3)
PRIN4990
DIMENSION FORMR ( 12),RPRINT( 4),FORMT ( 12),ITT (20)
PRIN5000
DATA FORMR /4H(13,, 4H12X,, 9*4H13X,, 1H)/
PRIN5010
DATA RPRINT /6HI6,I6,, 6HI7,I6,, 6HI8,4X,, 6HI9,4X,/
PRIN5020
DIMENSION NUMBER( 25)
PRIN5030
DATA NUMBER /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
PRIN5040
1
20,21,22,23,24,25/
PRIN5050

```

I=0	PRIN5060
DO 311 M=1,NGRID	PRIN5070
IF (NGAP(M).EQ.0) GO TO 311	PRIN5080
I=I+1	PRIN5090
WRITE (6,312) PLABL,M	PRIN5100
312 FORMAT (1H0,53X,A6,2H (,I2,11H) GRIDPLANE,/)	PRIN5110
NUM2=0	PRIN5120
292 NUM1=NUM2+1	PRIN5130
NUM2=NUM2+10	PRIN5140
IF (NUM2.GT.ICROSS) NUM2=ICROSS	PRIN5150
WRITE (6,293) (NUMBER(L), L=NUM1,NUM2)	PRIN5160
293 FORMAT (1H0,110,9I13)	PRIN5170
WRITE (6,294)	PRIN5180
294 FORMAT (1H0)	PRIN5190
DO 310 K=1,INDOWN	PRIN5200
N=0	PRIN5210
NCOUNT=1	PRIN5220
DO 301 MM=1,12	PRIN5230
301 FORMT(MM)=FORMR(MM)	PRIN5240
DO 300 J=NUM1,NUM2	PRIN5250
NCOUNT=NCOUNT+1	PRIN5260
IT1=BBTH(I,J,K)-459.5	PRIN5270
IF (IT1.LE.-450) GO TO 300	PRIN5280
IT2=BTTL(I,J,K)-459.5	PRIN5290
IF (IT1.EQ.IT2 .AND. GAP (I,J,K).LT.0.0) GO TO 302	PRIN5300
N=N+2	PRIN5310
ITT(N-1)=IT1	PRIN5320
ITT(N)=IT2	PRIN5330
FORMT(NCOUNT)=RPRINT(2)	PRIN5340
IF (NCOUNT.EQ.2) FORMT(NCOUNT)=RPRINT(1)	PRIN5350
GO TO 300	PRIN5360
302 N=N+1	PRIN5370
ITT(N)=IT1	PRIN5380
FORMT(NCOUNT)=RPRINT(4)	PRIN5390
IF (NCOUNT.EQ.2) FORMT(NCOUNT)=RPRINT(3)	PRIN5400
300 CONTINUE	PRIN5410
IF (N.EQ.0) GO TO 309	PRIN5420
WRITE (6,FORMT) K,(ITT(L),L=1,N)	PRIN5430
GO TO 310	PRIN5440
309 WRITE (6,FORMT) K	PRIN5450
310 CONTINUE	PRIN5460
IF (NUM2.LT.ICROSS) GO TO 292	PRIN5470
311 CONTINUE	PRIN5480
RETURN	PRIN5490
END	PRIN5500


```

SUBROUTINE ARRAY (IX,X,IFY)
INCLUDE COMDIM
=====ARRA 10
CB PRINT A THREE-DIMENSIONAL ARRAY
=====ARRA 20
C=====ARRA 30
C=====ARRA 40
C=====ARRA 50
C=====ARRA 60
DIMENSION NUMBER( 50),X ( IQ,JQ,KQ),IX ( IQ,JQ,KQ)
DATA NUMBER /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
- 20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,
- 36,37,38,39,40,41,42,43,44,45,46,47,48,49,50/
=====ARRA 70
C=====ARRA 80
C=====ARRA 90
C=====ARRA 100
C WRITE THE PROPER HEADING
NADD=10
GO TO (10,20,30), NPRINT
10 WRITE (6,1) PNAME(3), PNAME(2)
1 FORMAT (1H0,42X,4HTHE ,A6,27H(K) DIRECTION IS HORIZONTAL/43X,
-4HTHE ,A6,25H(J) DIRECTION IS VERTICAL)
MM=IMAX
NK=KMAX
GO TO 40
20 WRITE (6,2) PNAME(1), PNAME(3)
2 FORMAT (1H0,42X,4HTHE ,A6,27H(I) DIRECTION IS HORIZONTAL/43X,
-4HTHE ,A6,25H(K) DIRECTION IS VERTICAL)
MM=JMAX
NI=IMAX
GO TO 40
30 WRITE (6,3) PNAME(1), PNAME(2)
3 FORMAT (1H0,42X,4HTHE ,A6,27H(I) DIRECTION IS HORIZONTAL/43X,
-4HTHE ,A6,25H(J) DIRECTION IS VERTICAL)
MM=KMAX
NI=IMAX
GO TO 40
40 DO 600 M=1,MM
NUM2=0
100 NUM1=NUM2+1
IF (IFY.EQ.1) NADD=25
NUM2=NUM2+NADD
IF(NUM2 .GT. N) NUM2=N
IF (NUM1 .EQ. 1) WRITE (6,4) PNAME(NPRINT), M
4 FORMAT(1H0,/,48X,A6,2H (,I2,7H) PLANE)
IF (IFY .EQ. 1) GO TO 219
WRITE(6,220) (NUMBER(L) ,L=NUM1,NUM2)
220 FORMAT(1H0,I11,9I12)
GO TO 221
219 WRITE (6,225) (NUMBER(L),L=NUM1,NUM2)
225 FORMAT (1H0,5X,25I5)
=====ARRA 110
=====ARRA 120
=====ARRA 130
=====ARRA 140
=====ARRA 150
=====ARRA 160
=====ARRA 170
=====ARRA 180
=====ARRA 190
=====ARRA 200
=====ARRA 210
=====ARRA 220
=====ARRA 230
=====ARRA 240
=====ARRA 250
=====ARRA 260
=====ARRA 270
=====ARRA 280
=====ARRA 290
=====ARRA 300
=====ARRA 310
=====ARRA 320
=====ARRA 330
=====ARRA 340
=====ARRA 350
=====ARRA 360
=====ARRA 370
=====ARRA 380
=====ARRA 390
=====ARRA 400
=====ARRA 410
=====ARRA 420
=====ARRA 430
=====ARRA 440
=====ARRA 450

```

```

221 WRITE (6,170) ARR A 460
170 FORMAT (1H0) ARR A 470
GO TO (200,300,400),NPRINT ARR A 480
C=====ARR A 490
C OUTPUT PRINT IN RADIAL-X DIRECTION, THETA-Y DIRECTION HORIZONTAL, ARR A 500
C AXIAL-Z DIRECTION VERTICAL ARR A 510
C=====ARR A 520
200 DO 420 J=1,JMAX ARR A 530
GO TO (421,422,423),IFY ARR A 540
421 WRITE (6,203) J,(IX(M,J,K),K=NUM1,NUM2) ARR A 550
203 FORMAT (I4,1X,I6,2415) ARR A 560
GO TO 420 ARR A 570
422 WRITE (6,201) J,(X(M,J,K),K=NUM1,NUM2) ARR A 580
201 FORMAT (I4,1P10E12.4) ARR A 590
GO TO 420 ARR A 600
423 WRITE (6,202) J,(X(M,J,K),K=NUM1,NUM2) ARR A 610
202 FORMAT (I4,1P10E12.5) ARR A 620
420 CONTINUE ARR A 630
IF(NUM2 .LT. N) GO TO 100 ARR A 640
GO TO 600 ARR A 650
C=====ARR A 660
C OUTPUT PRINT IN AXIAL-Z DIRECTION, RADIAL-X DIRECTION HORIZONTAL, ARR A 670
C THETA-Y DIRECTION VERTICAL ARR A 680
C=====ARR A 690
300 DO 320 K=1,KMAX ARR A 700
GO TO (321,322,323),IFY ARR A 710
321 WRITE (6,203) K,(IX(I,M,K), I=NUM1,NUM2) ARR A 720
GO TO 320 ARR A 730
322 WRITE (6,201) K,(X(I,M,K), I=NUM1,NUM2) ARR A 740
GO TO 320 ARR A 750
323 WRITE (6,202) K,(X(I,M,K), I=NUM1,NUM2) ARR A 760
320 CONTINUE ARR A 770
IF(NUM2 .LT. N) GO TO 100 ARR A 780
GO TO 600 ARR A 790
C=====ARR A 800
C OUTPUT PRINT IN THETA-Y DIRECTION, RADIAL-X DIRECTION HORIZONTAL, ARR A 810
C AXIAL-Z DIRECTION VERTICAL ARR A 820
C=====ARR A 830
400 DO 230 J=1,JMAX ARR A 840
GO TO (231,232,233), IFY ARR A 850
231 WRITE (6,203) J,(IX(I,J,M),I=NUM1,NUM2) ARR A 860
GO TO 230 ARR A 870
232 WRITE (6,201) J,( X(I,J,M),I=NUM1,NUM2) ARR A 880
GO TO 230 ARR A 890
233 WRITE (6,202) J,( X(I,J,M),I=NUM1,NUM2) ARR A 900
230 CONTINUE ARR A 910
IF(NUM2 .LT. N) GO TO 100 ARR A 920
600 CONTINUE ARR A 930
RETURN ARR A 940
C=====ARR A 950
END ARR A 960

```

```

SUBROUTINE CUSTOM                                CUST 10
  INCLUDE      COMDIM                            CUST 20
C=====                                        CUST 30
C  DO WHAT THE USER WANTS TO DO                CUST 40
C=====                                        CUST 50
C=====                                        CUST 60
C=====                                        CUST 70
C=====                                        CUST 80
C=====                                        CUST 90
C=====                                        CUST 100
  RETURN                                         CUST 110
  END                                             CUST 120

```

```

SUBROUTINE PUN
INCLUDE      COMDIM
=====
C PUNCH THE FINAL TEMPERATURE DISTRIBUTION.
C
C DIMENSION      ALABEL(4), TPUN(1)
EQUIVALENCE    (TT ,TPUN )
DATA ALABEL    /6HHEADER, 5HTEMP , 5HINLET, 5HOUTLT/
=====
C
C
C WRITE (6,210) (ZA(I),I=1,12)
IF(.NOT.SW(10)) GO TO 300
CURTI=0.0
NITER=0
300 BIIMAX=IMAX
    BJMAX=JMAX
    BKMAX=KMAX
    BNITER=NBITER
    WRITE (6,220) CURTI,BIIMAX,BJMAX,BKMAX,BNITER,ALABEL(1)
    PUNCH 230, CURTI,BIIMAX,BJMAX,BKMAX,BNITER,ALABEL(1)
    NT=0
    DO 240 K=1,KMAX
    DO 240 J=1,JMAX
    DO 240 I=1,IMAX
    NT=NT+1
    TPUN(NT)=T(I,J,K)
240 CONTINUE
    CALL PUNCHY (TPUN,NT,ALABEL(2))
    CALL PUNCHY (TI,MAXFLO,ALABEL(3))
    CALL PUNCHY (TO,MAXFLO,ALABEL(4))
    PUNCH 250
    WRITE (6,260)
    RETURN
=====
C=
210 FORMAT (1H1,30X,12A6,///,40X,34HPUNCHED TEMPERATURE DECK (R)
1,/)
220 FORMAT (10X,E12.6,4F12.1,12X,A6)
230 FORMAT (E12.6,4F12.1,12X,A6)
250 FORMAT (///)
260 FORMAT (1H0,20X,52H(REMOVE THE FIVE BLANK CARDS AT END OF PUNCHED
1)DECK))
=====
C=
CE *****
CE *****
CE *****
CE *****
=====

```

```

SUBROUTINE PUNCHY (TP,LENGTH,ANAME)                                PUN 460
C= =====PUN 470
C PUNCH THE TEMPERATURE DISTRIBUTION DECK                        PUN 480
C= =====PUN 490
C= =====PUN 500
DIMENSION TP ( 1),FORM ( 9),FORMT ( 9),FB ( 3) PUN 510
DATA FORMT /6H , 6H 1P, 6*6HE12.6,, 6HA5,I3)/ PUN 520
DATA FB /6H( 10X,, 6H( , 6H 12X,/ PUN 530
C= =====PUN 540
N=LENGTH/6 PUN 550
NLEFT=LENGTH-N*6 PUN 560
NN1=1 PUN 570
NN2=6 PUN 580
DO 90 I=1,9 PUN 590
FORM(I)=FORMT(I) PUN 600
90 CONTINUE PUN 610
DO 100 KK=1,N PUN 620
FORM(1)=FB(1) PUN 630
WRITE (6,FORM) (TP(I),I=NN1,NN2),ANAME,KK PUN 640
FORM(1)=FB(2) PUN 650
PUNCH FORM, (TP(I),I=NN1,NN2),ANAME,KK PUN 660
NN1=NN2+1 PUN 670
NN2=NN2+6 PUN 680
100 CONTINUE PUN 690
C= =====PUN 700
IF (NLEFT.EQ.0) GO TO 120 PUN 710
KK=..+1 PUN 720
J=NLEFT+3 PUN 730
DO 110 I=J,8 PUN 740
FORM(1)=FB(3) PUN 750
110 CONTINUE PUN 760
FORM(1)=FB(1) PUN 770
WRITE (6,FORM) (TP(I),I=NN1,LENGTH),ANAME,KK PUN 780
FORM(1)=FB(2) PUN 790
PUNCH FORM, (TP(I),I=NN1,LENGTH),ANAME,KK PUN 800
C= =====PUN 810
120 RETURN PUN 820
END PUN 830

```

REFERENCES

1. Petersen, J. F., "TAC3D, a General Purpose Three-Dimensional Heat Transfer Computer Code - User's Manual," USAEC Report GA-9263, Gulf General Atomic, September, 1969.
2. Clark, S. S. and Petersen, J. F., "TAC2D, a General Purpose Two-Dimensional Heat Transfer Code - Mathematical Formulations and Programmer's Guide," USAEC Report GA-9262, Gulf General Atomic, September, 1969.
3. Petersen, J. F., "TAC2D, a General Purpose Two-Dimensional Heat Transfer Computer Code - User's Manual," USAEC Report GA-8868, Gulf General Atomic, September, 1969.
4. Douglas, J., "Alternating Direction Methods for Three Space Variables," Numer. Math. 4, 41-63 (1962).
5. Gross, S., "An Improved Finite Difference Method of Heat Transfer Calculations," J. Spacecraft, Vol. 4, No. 4, p. 538, 1967.
6. Carslaw, H. S. and Jaeger, J. C., Conduction of Heat in Solids, Oxford at the Clarendon Press, p. 184, 1959.
7. Handbook of Mathematical Functions, M. Abramowitz and I. A. Stegun, Eds., Dover Publications, 1965.
8. Schneider, P. J., Temperature Response Charts, John Wiley and Sons, Inc., 1963.