2 cy.

Gulf General Atomic
Incorporated

AEC RESEARCH AND DEVELOPMENT REPORT

GA-9264
[redacted]

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

LEGAL NOTICE

[redacted]

DISCLAIMER

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

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.

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

Table of Contents (Continued)

LIST OF TABLES

LIST OF FIGURES

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-ft^3

ρ is density, lb/ft^3

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.

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.

Model Envelope	Coordinate Axes
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 θ = 0° and θ = 360°.

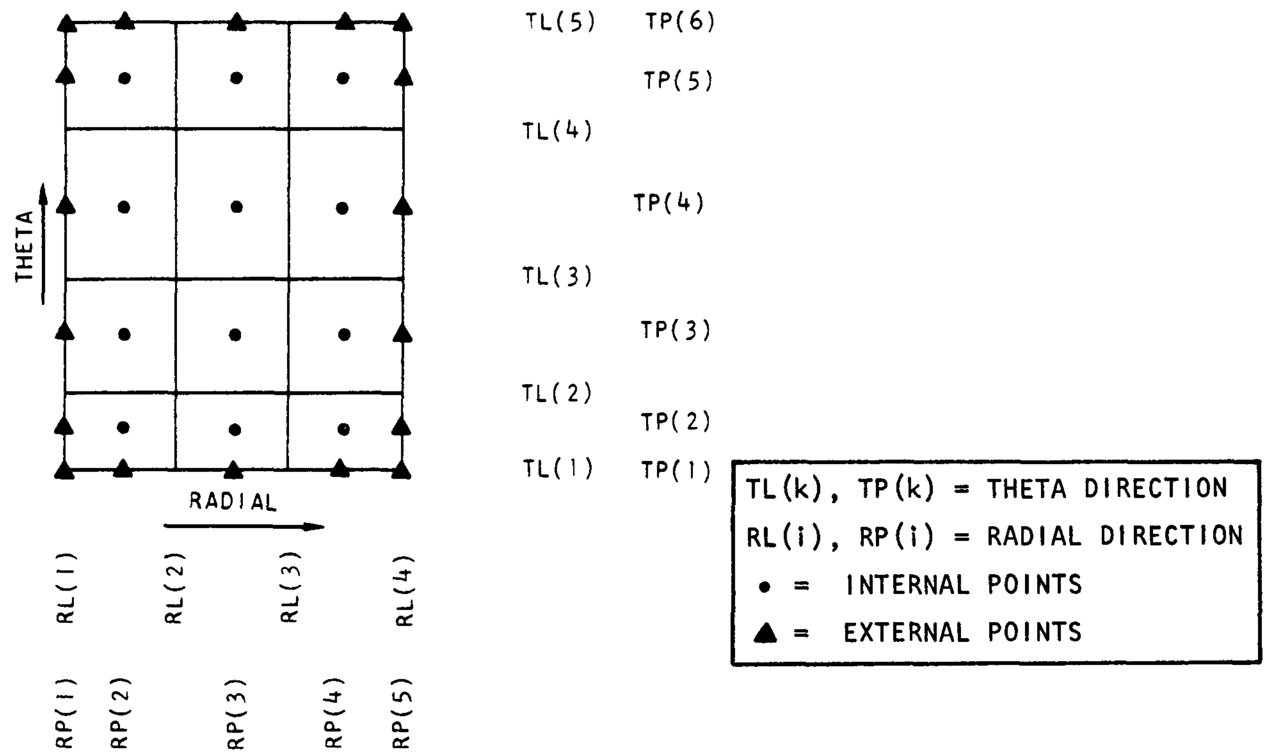


Fig. 1. Axial plane of a typical rectangular geometry envelope showing grid planes and mesh points

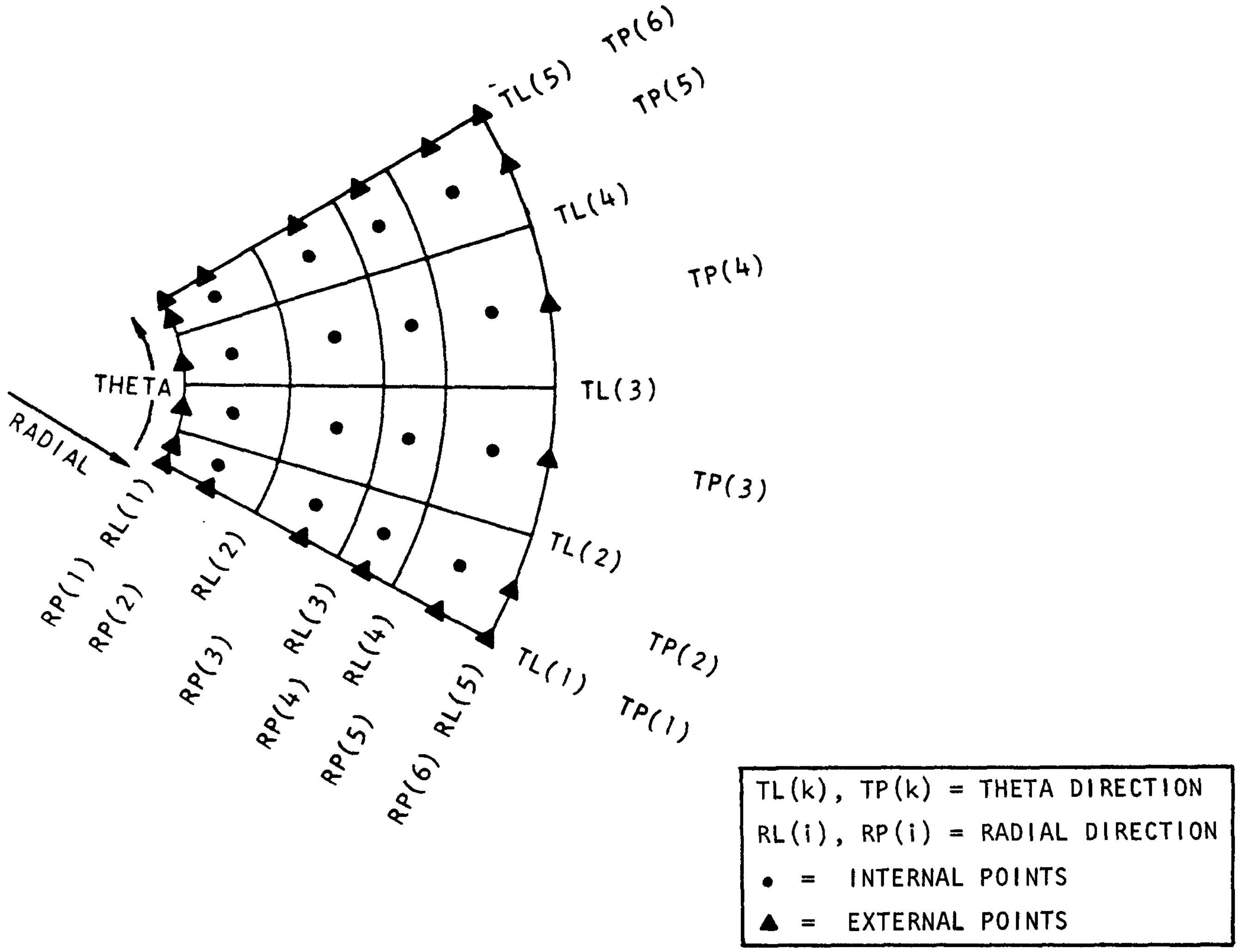


Fig. 2. Axial plane of a typical cylindrical geometry envelope showing grid planes and mesh points

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.

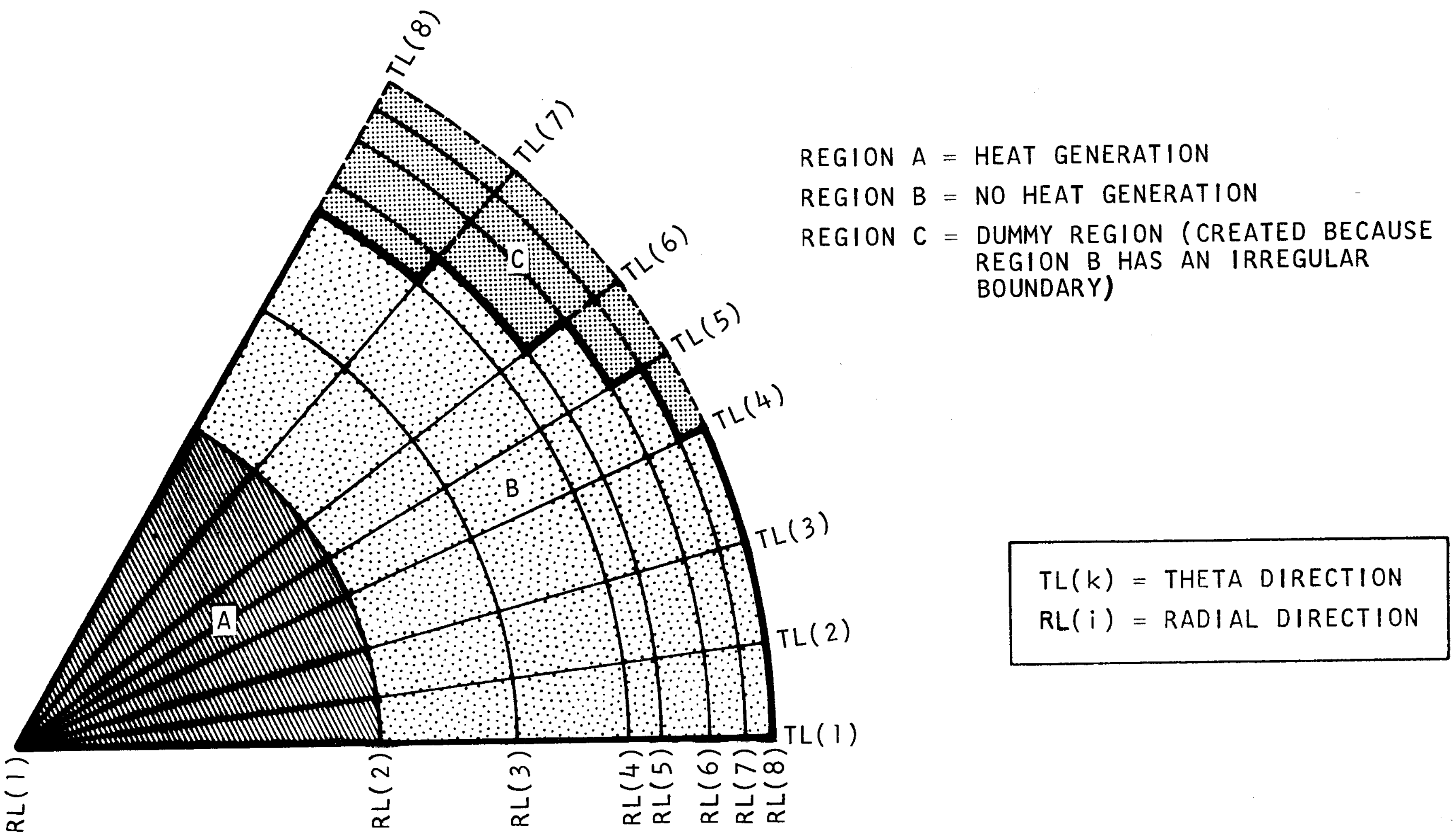


Fig. 3. Axial plane of a typical problem model in cylindrical geometry

TABLE 2

INTERNAL BLOCK DEFINITION

	Block Grid Plane Boundaries		
Block Number	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 Grid Plane Boundaries		
Block Number (Boundary)	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 subprograms. 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)     .
```

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.

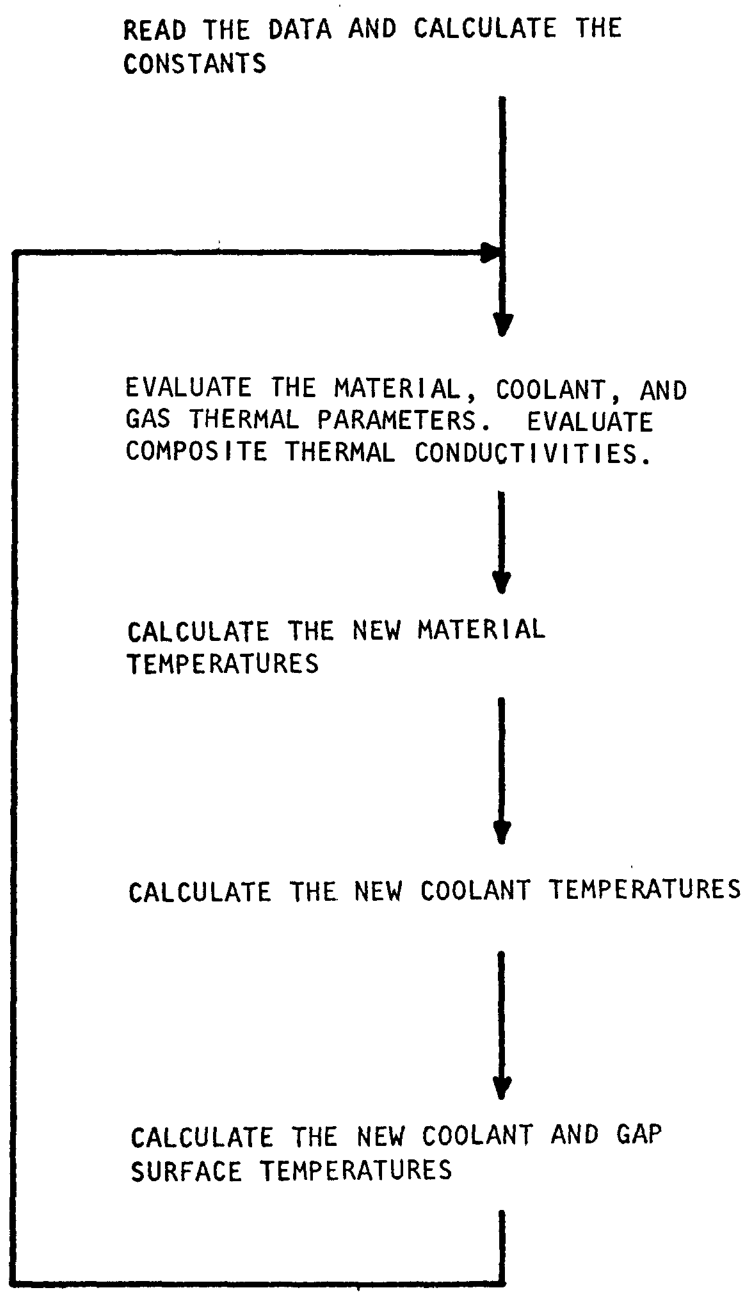


Fig. 4. General flow of logic in TAC3D

APPENDIX A

MATHEMATICAL SOLUTION

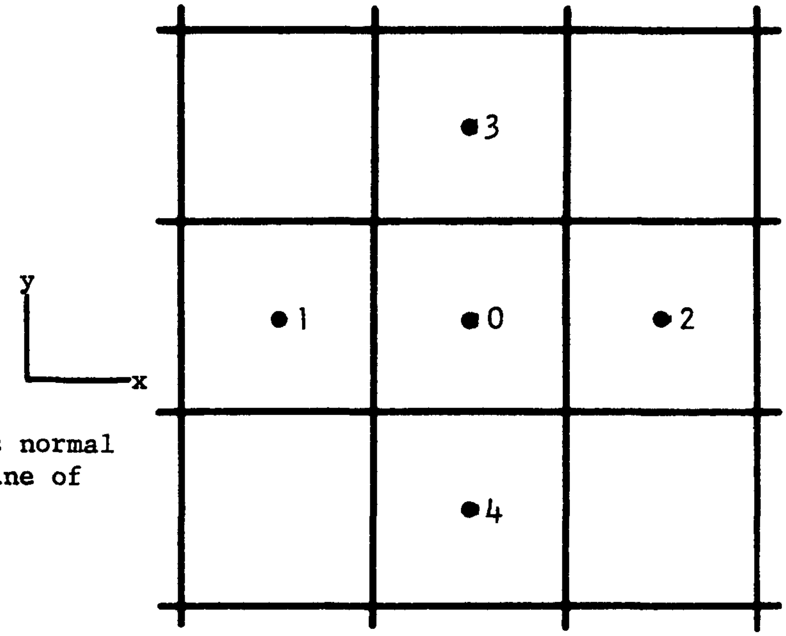
The time dependent heat conduction differential equation is

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

where

- k = thermal conductivity in each of the 3 dimensions
- T = temperature (°R)
- S = heat generation (Btu/ft^3-hr)
- ρc = volumetric specific heat (Btu/ft^3-°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.



z-axis normal to plane of 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

$$\frac{K_1A_1}{\Delta X_1}(T_1 - T_0) + \frac{K_2A_2}{\Delta X_2}(T_2 - T_0) + \frac{K_3A_3}{\Delta X_3}(T_3 - T_0)$$

$$+ \frac{K_4A_4}{\Delta X_4}(T_4 - T_0) + \frac{K_5A_5}{\Delta X_5}(T_5 - T_0) + \frac{K_6A_6}{\Delta X_6}(T_6 - T_0)$$

$$+ S_0V_0 = \frac{(T_0' - T_0)\rho c\, V_0}{\Delta\tau} \qquad \text{(A-2)}$$

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 (°R)

T'_0 = temperature of point 0 (°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:

$$C_1\left(\frac{(T'_1 + T_1)}{2} - T_0\right) + C_2\left(\frac{(T'_2 + T_2)}{2} - T_0\right) + C_3(T_3 - T_0) + C_4(T_4 - T_0)$$

$$+ C_5(T_5 - T_0) + C_6(T_6 - T_0) = C_7(T'_0 - T_0) - W \qquad \text{(A-3)}$$

and

$$C_1\left(\frac{(T_1' + T_1)}{2} - T_0\right) + C_2\left(\frac{(T_2' + T_2)}{2} - T_0\right) + C_3\ \frac{(T_3'' + T_3)}{2} - T_0 \qquad \text{(A-4)}$$

$$+ C_4\left(\frac{(T_4'' + T_4)}{2} - T_0\right) + C_5(T_5 - T_0) + C_6(T_6 - T_0) = C_7(T_0'' - T_0) - W$$

and

$$C_1\left(\frac{(T_1' + T_1)}{2} - T_0\right) + C_2\left(\frac{(T_2' + T_2)}{2} - T_0\right) + C_3\left(\frac{(T_3'' + T_3)}{2} - T_0\right)$$

$$+ C_4\left(\frac{(T_4'' + T_4)}{2} - T_0\right) + C_5\left(\frac{(T_5''' + T_5)}{2} - T_0\right) + C_6\left(\frac{(T_6''' + T_6)}{2} - T_0\right)$$

$$= C_7(T_0''' + T_0) - W \qquad \text{(A-5)}$$

where $C_i = K_iA_i/\Delta x_i$, $W = S_oV_o$ and $C_7 = \rho cV_o/\delta\tau$.

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;

$$C_3\ \frac{(T_3'' - T_3)}{2} + C_4\ \frac{(T_4'' - T_4)}{2} = C_7(T_0'' - T_0') \qquad \text{(A-6)}$$

$$C_5\ \frac{(T_5''' - T_5)}{2} + C_6\ \frac{(T_6''' - T_6)}{2} = C_7(T_0''' - T_0''). \qquad \text{(A-7)}$$

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

$$\frac{C_1T_1'}{2} - C_7T_0' + \frac{C_2T_2'}{2} = - C_7T_0 - C_1\left(\frac{T_1}{2} - T_0\right) - C_2\left(\frac{T_2}{2} - T_0\right)$$

$$- C_3(T_3 - T_0) - C_4(T_4 - T_0) - C_5(T_5 - T_0) - C_6(T_6 - T_0) - W$$

or

$$AT_1' + BT_0' + CT_2' = D, \qquad \text{(A-8)}$$

$$\frac{C_3T_3''}{2} - C_7T_0'' + \frac{C_4T_4''}{2} = - C_7T_0' + \frac{C_3T_3}{2} + \frac{C_4T_4}{2}$$

or

$$AT_3'' + BT_0'' + CT_4'' = D \qquad \text{(A-9)}$$

and

$$\frac{C_5T_5'''}{2} - C_7T_0''' + \frac{C_6T_6'''}{2} = - C_7T_0'' + \frac{C_5T_5}{2} + \frac{C_6T_6}{2}$$

or

$$AT_5''' + BT_0''' + CT_6''' = D \qquad \text{(A-10)}$$

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

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

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

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

In the Axial direction,

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

and

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

In the theta direction,

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

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

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

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

	Grid Plane	Point	Gap Thickness
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,

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

$$\times \left[TL(K) - TL(K-1) - \frac{\Delta g_T(I,J) + \Delta g_T(I+1,J)}{2.0}\right] \qquad (B-5)$$

$$RZ(I,J,K) = \left\{\left[RL(I) - \frac{\Delta g_R(J,K) + \Delta g_R(J + 1,K)}{2.0}\right]^2 - \left[RL(I-1)\right]^2\right\}$$

$$\times\left[\frac{TL(K) - TL(K-1) - \left\{\frac{1}{2.0}\left[\Delta g_T(I,J) + \Delta g_T(I,J+1)\right]\right\}}{ZP(J+1) - ZP(J)}\right] \qquad (B-6)$$

$$RT(I,J,K) = \left[RL(I) - RL(I-1) - \frac{\Delta g_R(J,K) + \Delta g_R(J,K+1)}{2.0}\right]$$

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

and

$$V(I,J,K) = \left\{\left\{[RL(I)-\Delta g_R(J,K)]^2\right\}-\left[RL(I-1)\right]^2\right\}[ZL(J) - ZL(J-1)-\Delta g_Z(I,K)]$$

$$\times\ [TL(K) - TL(K-1)-\Delta g_T(I,J)](0.5). \qquad (B-8)$$

For the rectangular geometry,

$$RR(I,J,K) = TL(K) - TL(K-1) - \frac{\Delta g_T(I,J) + \Delta g_T(I+1,J)}{2.0}$$

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

$$RZ(I,J,K) = TL(K) - TL(K-1) - \frac{\Delta g_T(I,J) + \Delta g_T(I,J+1)}{2.0}$$

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

$$RT(I,J,K) = RL(I) - RL(I-1) - \frac{\Delta g_R(J,K) + \Delta g_R(J,K+1)}{2.0}$$

$$\times \frac{ZL(J) - ZL(J-1) - \left\{\frac{1}{2.0}\left[\Delta g_Z(I,K) + \Delta g_Z(I,K+1)\right]\right\}}{TP(K+1) - TP(K)} \qquad \text{(B-11)}$$

and

$$V(I,J,K) = [TL(K) - TL(K-1) - \Delta g_T(I,J)][RL(I) - RL(I-1) - \Delta g_R(J,K)]$$

$$\times [ZL(J) - ZL(J-1) - \Delta g_Z(I,K)] \qquad \text{(B-12)}$$

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.

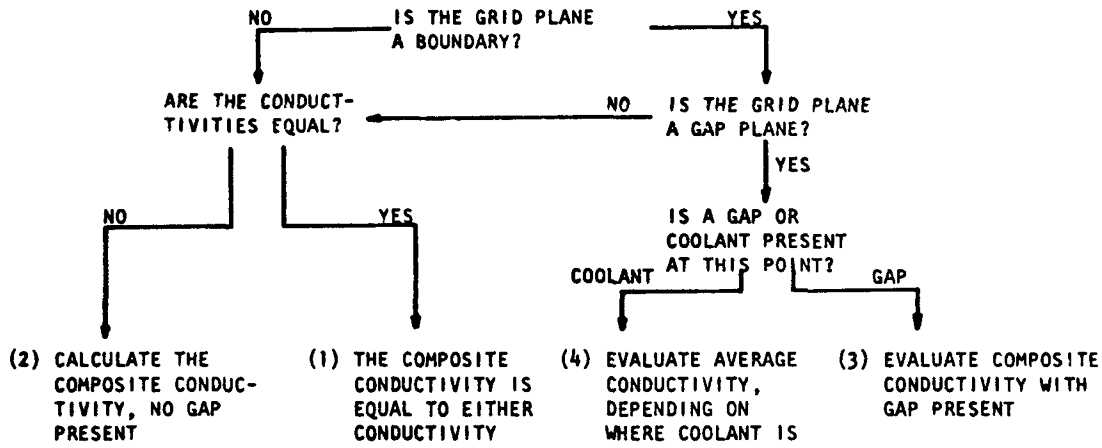


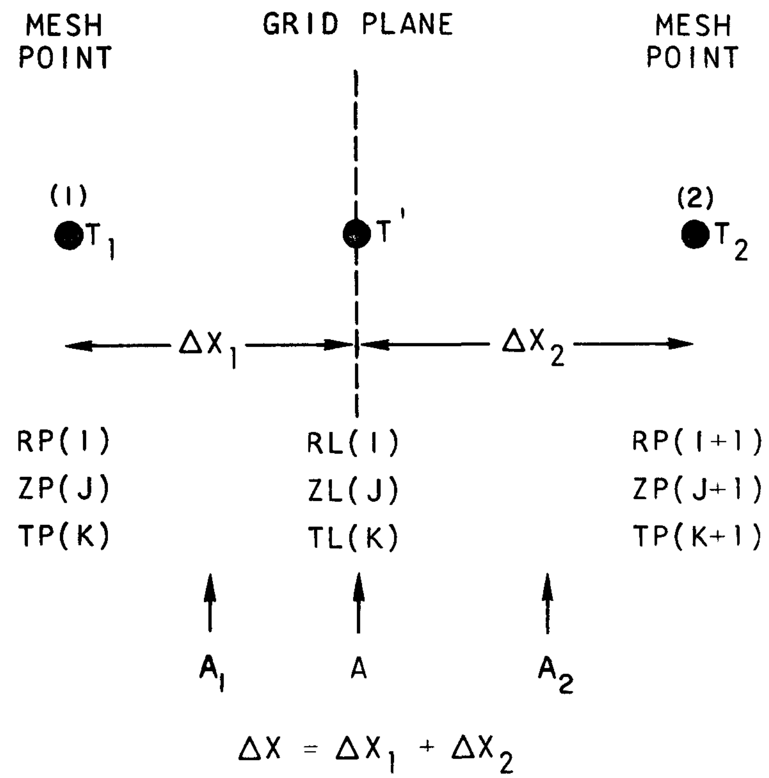
Fig. C.1. Logical search at each grid plane

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

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

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

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

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

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

Now, the equation for K may be rewritten as

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

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,

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

and, in the radial direction for rectangular geometry,

$$A = A_1 = A_2$$
$$\Delta X = RP(I + 1) - RP(I) \tag{C-6}$$
$$\Delta X_1 = RL(I) - RP(I) \quad .$$

In the theta direction for cylindrical geometry,

$$A = A_1 = A_2$$
$$\Delta X = [TP(K + 1) - TP(K)] \; [RP(I)] \tag{C-7}$$
$$\Delta X_1 = [TL(K) - TP(K)] \; [RP(I)] \quad .$$

and, in the theta direction for rectangular geometry,

$$A = A_1 = A_2$$
$$\Delta X = TP(K + 1) - TP(K) \tag{C-8}$$
$$\Delta X_1 = TL(K) - TP(K) \quad .$$

In the axial direction for rectangular and cylindrical geometries,

$$A = A_1 = A_2$$

$$\Delta X = ZP(J + 1) - ZP(J) \qquad (C-9)$$

$$\Delta X_1 = ZL(J) - ZP(J) \quad .$$

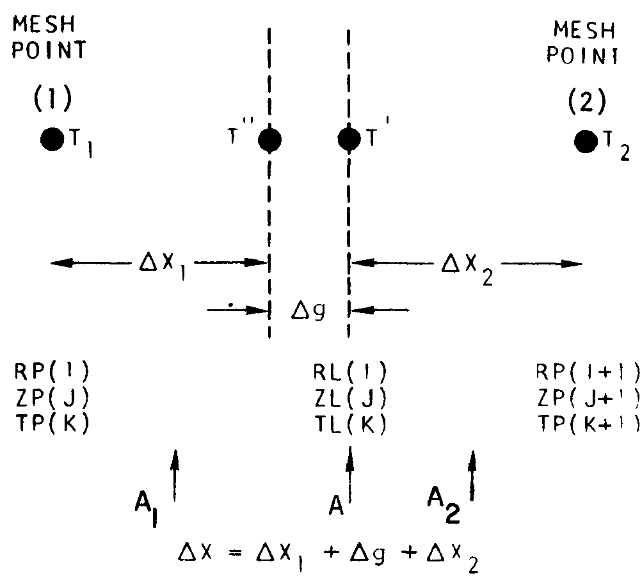
The nomenclature and definition of the geometrical constant $(\Delta X_1/A_1)/(\Delta X/A)$ in the coordinate directions for the two available geometry types are given in Table C.1.

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

$$\frac{KA(T_1 - T_2)}{\Delta X} \equiv \frac{k_1A_1(T_1 - T'')}{\Delta X_1} = \frac{k_2A_2(T' - T_2)}{\Delta X_2} = \frac{k_gA_g(T'' - T')}{\Delta g}$$

$$+ \frac{\sigma A_g(T''^4 - T'^4)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} \qquad \text{(C-10)}$$

where σ is the Stefan-Boltzmann constant ($.1713 \times 10^{-8}$ Btu/hr-ft^2-°R^4) 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

$$\frac{k_1A_1}{\Delta X_1}(T_1 - T'') = \frac{k_2A_2}{\Delta X_2}(T' - T_2) \quad . \qquad \text{(C-11)}$$

When solving for T", one obtains

$$T'' = \frac{k_2A_2\Delta X_1}{k_1A_1\Delta X_2}(T_2 - T') + T_1 \equiv B(T_2 - T') + T_1 \quad , \text{(C-12)}$$

$$\frac{k_2A_2}{\Delta X_2}(T' - T_2) = \frac{k_gA_g}{\Delta g}(T'' - T')\left[1 + \frac{\sigma\Delta g(T''^2 + T'^2)(T'' + T')}{k_g\left(\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1\right)}\right] \text{(C-13)}$$

and

$$T' - T_2 = \frac{k_gA_g\Delta X_2}{k_2A_2\Delta g}(T'' - T')\,G \equiv H(T'' - T') \quad . \text{(C-14)}$$

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

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

and

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

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 $Ag/(A_2/\Delta X_2)$ -- 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,

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

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

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

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

and

$$\Delta g = \Delta g_R \quad . \tag{C-17}$$

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

$$\Delta g = \Delta g_R \quad . \tag{C-18}$$

In the axial direction for rectangular and cylindrical geometries,

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

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

$$\Delta X_1 = ZL(J) - ZP(J) - \Delta g_Z \quad ,$$

$$\Delta X_2 = ZP(J + 1) - ZL(J) \quad , \qquad \text{(C-19)}$$

and

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

In the theta direction for cylindrical geometry,

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

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

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

$$\Delta X_2 = [TP(K + 1) - TL(K)][RP(I)] \quad ,$$

and

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

In the theta direction for rectangular geometry,

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

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

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

$$\Delta X_2 = TP(K + 1) - TL(K) \quad ,$$

and

$$\Delta g = \Delta g_T \quad . \qquad \text{(C-21)}$$

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 $\ln\left[\frac{RL(IHS)-RDG(L)}{RP(IHS)}\right] \Big/ \ln\left[\frac{RP(IHS+1)}{RL(IHS)}\right]$ 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 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 Rectangular $\frac{TL(KHS)-TP(KHS)-TDG(L)}{TP(KHS+1)-TL(KHS)}$

Table C.4

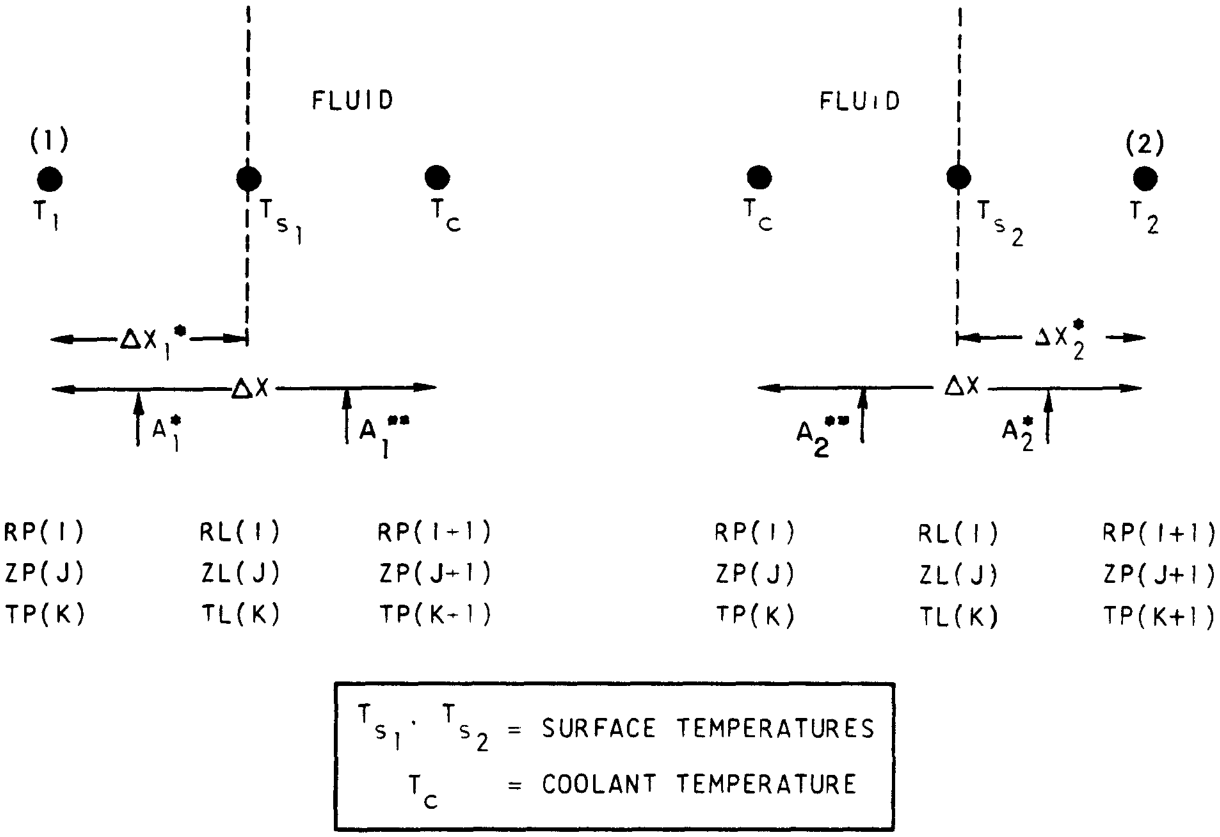
GEOMETRICAL CONSTANT

$Ag/(A_2/\Delta X_2)$

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.



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

$$\frac{k_1 A_1^*}{\Delta X_1^*}\left(T_1 - T_{s_1}\right) = h_1 A_1\left(T_{s_1} - T_c\right) + \frac{\sigma A_1\left(T_{s_1}^4 - T_{s_2}^4\right)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1}$$

$$\doteq \frac{K_1 A_1^{**}\left(T_1 - T_c\right)}{\Delta X} , \qquad \text{(C-22)}$$

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-ft^2-°R^4

$\varepsilon_1, \varepsilon_2$ = thermal emissivities.

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

$$\frac{\sigma A_1\left(T_{s_1}^4 - T_{s_2}^4\right)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} \qquad \text{(C-23)}$$

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

A composite heat transfer coefficient H_1 is defined by

$$h_1A_1\left(T_{s_1} - T_c\right) + \frac{\sigma A_1\left(T_{s_1}^4 - T_{s_2}^4\right)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} \equiv H_1A_1\left(T_{s_1} - T_c\right). \quad \text{(C-24)}$$

Let

$$\frac{\sigma\left(T_{s_1}^4 - T_{s_2}^4\right)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} = \text{RAD} \quad . \quad \text{(C-25)}$$

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

$$\frac{k_1A_1^*}{\Delta X_1^*}\left(T_1 - T_{s_1}\right) = H_1A_1\left(T_{s_1} - T_c\right) \quad , \quad \text{(C-26)}$$

and

$$T_{s_1} = \frac{H_1A_1T_c + \frac{k_1A_1^*}{\Delta X_1^*}T_1}{H_1A_1 + \frac{k_1A_1^*}{\Delta X_1^*}} = \frac{\frac{1}{k_1}T_c + \frac{1}{H_1}\frac{A_1^*/\Delta X_1^*}{A_1}T_1}{\frac{1}{k_1} + \frac{1}{H_1}\frac{A_1^*/\Delta X_1^*}{A_1}} \quad . \text{(C-27)}$$

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

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

Rewriting (C-22) as

$$\frac{K_1A_1^{**}\ (T_1 - T_c)}{\Delta X} = H_1A_1\left(T_{s_1} - T_c\right) \quad , \quad \text{(C-29)}$$

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

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

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

$$\frac{k_2 A_2^*}{\Delta X_2^*}\left(T_2 - T_{s_2}\right) = h_2 A_2\left(T_{s_2} - T_c\right) + \frac{\sigma\, A_1\left(T_{s_2}^4 - T_{s_1}^4\right)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1}$$

$$= \frac{K_2 A_2^{**}(T_2 - T_c)}{\Delta X} \qquad \text{(C-31)}$$

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

$$h_2 A_2\left(T_{s_2} - T_c\right) - \frac{\sigma\, A_1\left(T_{s_1}^4 - T_{s_2}^4\right)}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} = H_2 A_2\left(T_{s_2} - T_c\right) \qquad \text{(C-32)}$$

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

$$T_{s_2} = \frac{\frac{1}{k_2} T_c + \frac{1}{H_2} \frac{A_2^*/\Delta X_2^*}{A_2} T_2}{\frac{1}{k_2} + \frac{1}{H_2} \frac{A_2^*/\Delta X_2^*}{A_2}} , \qquad \text{(C-33)}$$

and

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

and

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

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

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

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

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

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

$$\text{Radial} = \frac{\text{RATIOK(IG)}}{\text{RP(I + 1) - RP(I)}}$$

$$\text{Axial} = \frac{1.0}{\text{ZP(J + 1) - ZL(J)}}$$

$$\text{Theta} = \frac{1.0}{\text{[TP(K + 1) - TL(K)][RP(I)]}} \qquad \text{(C-37)}$$

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

$$\text{Radial} = \frac{1.0}{\text{RL(I)} \times \ln\left[\frac{\text{RP(I + 1)}}{\text{RP(I)}}\right]}$$

$$\text{Axial} = \frac{1.0}{\text{ZP(J + 1) - ZP(J)}}$$

$$\text{Theta} = \frac{1.0}{\text{[TP(K + 1) - TP(K)][RP(I)]}} \qquad \text{(C-38)}$$

For A_1/A_2,

$$\text{Radial} = \frac{\text{RL(ILS - 1)}}{\text{RL(IHS)}}$$

$$\text{Axial} = 1.0$$

$$\text{Theta} = 1.0 \qquad \text{(C-39)}$$

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, theppoints 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.

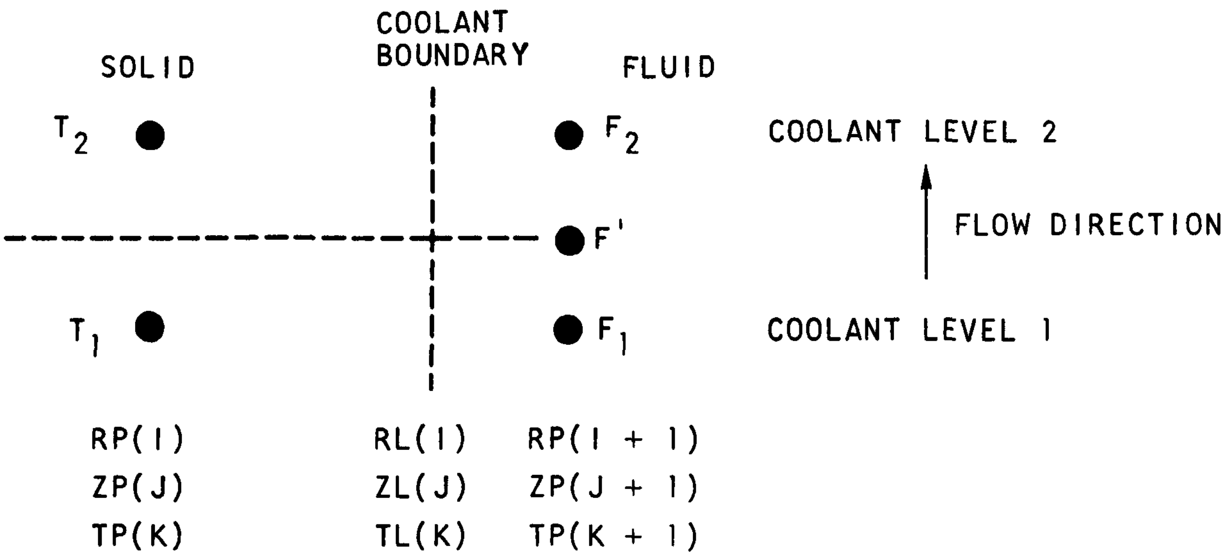
2. New coolant temperatures based on the results of the above solution are calculated.

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

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

where

dF = differential coolant temperature increment

h_i = heat transfer coefficient (Btu/ft^2-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

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

When

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

are defined, the equation becomes

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

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

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

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

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

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

and

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

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

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

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

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

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

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

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

When

$$P = \sum_i C_i T_i \text{ and } R = \sum_i C_i \quad ,$$

$$F_T = \frac{P}{R}\left(1 - e^{-R/WC_p}\right) + F_B e^{-R/WC_p} \quad . \qquad \text{(D-10)}$$

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

$$WC_p(F_T - F_B) = \sum_i C_i(T_i - F_{av})$$

$$= \sum_i C_i T_i - \sum_i C_i F_{av}$$

$$= P - RF_{av} \quad , \qquad \text{(D-11)}$$

and

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

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.

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

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

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

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

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

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/°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

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

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

$$0.0 < T_i < 1.0 \times 10^6 \quad °R$$

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

$$R_i = \frac{\dfrac{C_i\,(T_i - T_i')}{\Delta\tau_i'}}{\displaystyle\sum_{j=1}^{6} \left| K_{ji}\left(\frac{(T_i + T_i')}{2} - \frac{(T_j + T_j')}{2}\right)\right| + |W_i|} \qquad \text{(E-5)}$$

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

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

R_i values at points where, as previously discussed, the C_i have been set at 1.0×10^{12} Btu/°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 in.; b = 3.6 in.; c = 2.4 in.

k = 10 Btu/hr-ft-°F; ρc_p = 50 Btu/ft^3-°F

$h = 50$ Btu/hr-ft^2-°F

$T_i = 600°F$; $T_f - 100°F$

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

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

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

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

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

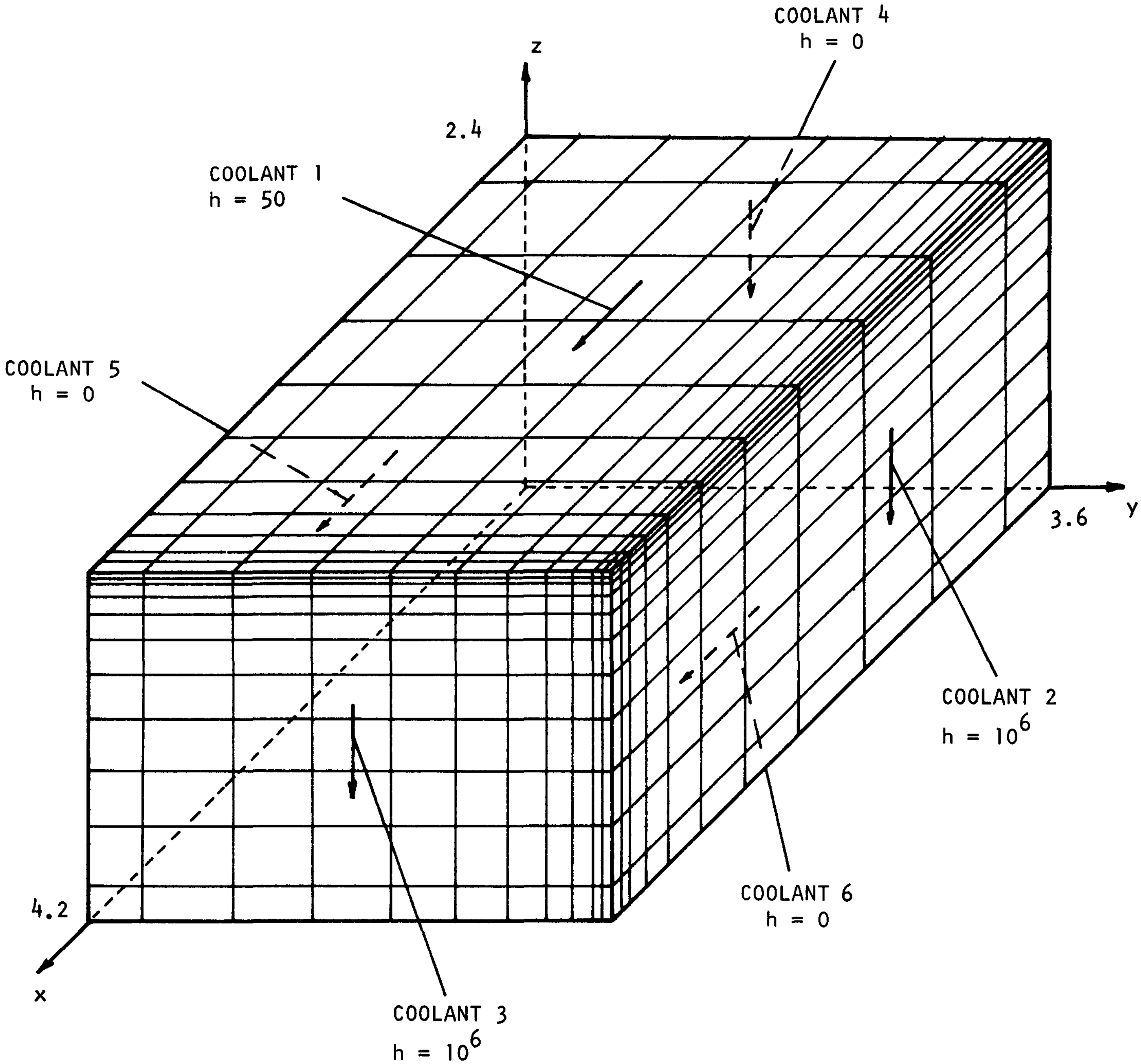


Fig. F.1. TAC3D sample problem — thermal model as specified by the input functions, are given in Table F.1. All coolants have the standard flow rate of $10^6 lb_m/hr$ and the standard specific heat of 1.0 Btu/lb_m-°F. Coolants 4, 5, and 6, which represent the adiabatic faces, have the standard heat transfer coefficient equal to 10^{-6} $Btu/hr-ft^2$-°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 DESCRIPTION

BLOCK NUMBER	. . . . BOUNDARIES							. . . . . GAPS					
	LOW X (INCHES)	HIGH X (INCHES)	LOW Y (INCHES)	HIGH Y (INCHES)	LOW Z (INCHES)	HIGH Z (INCHES)	MATERIAL	X (INCHES)	MATERIAL	Y (INCHES)	MATERIAL	Z (INCHES)	MATERIAL
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)

X BOUNDARY ASSIGNMENTS

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

TABLE F.2

(Continued)

Y BOUNDARY ASSIGNMENTS

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

TABLE F.2

(Continued)

Z BOUNDARY ASSIGNMENTS

SEQUENCE NUMBER	POINT LOCATION (INCHES)	GRID LINE LOCATION (INCHES)	COOLANT OR GAP BOUNDARY NUMBER
1	.0000		
1		.0000	1
2	.2400		
2		.4800	
3	.6600		
3		.8400	
4	1.0200		
4		1.2000	
5	1.3800		
5		1.5600	
6	1.6800		
6		1.8000	
7	1.9200		
7		2.0400	
8	2.1000		
8		2.1600	
9	2.2200		
9		2.2800	
10	2.3160		
10		2.3520	
11	2.3760		
11		2.4000	2
12	2.4000		

TIME HISTORY

END OF TIME PERIOD HOURS	MINUTES	SECONDS	TIME STEP HOURS	MINUTES	SECONDS	PRINT FREQUENCY
3.000000-05	1.800000-03	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

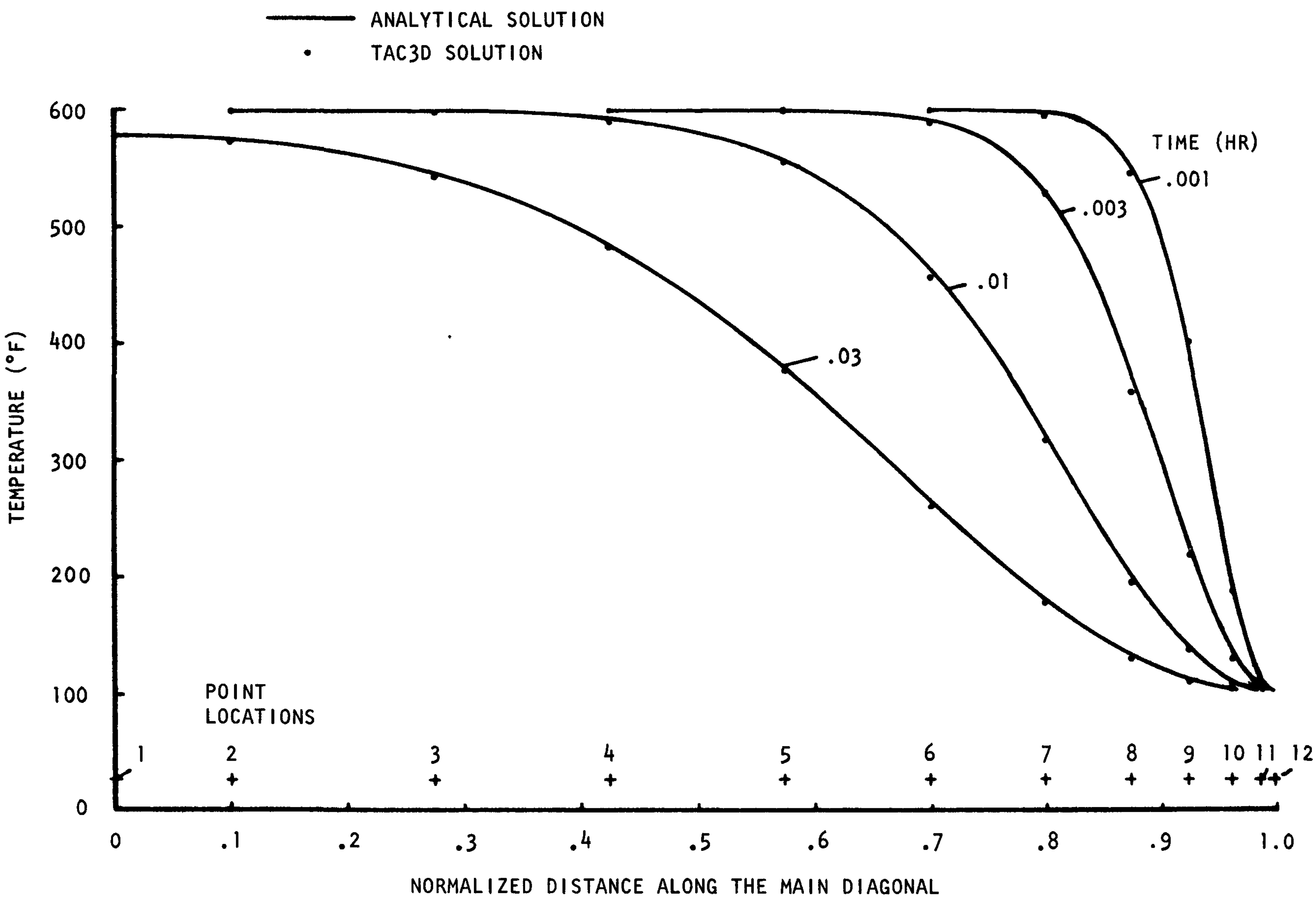


Fig. F.2. TAC3D sample problem — comparison of code solution with analytical solution

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
CONDUC

*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, CONDUC, 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, CONDUC

      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 CONDUC - 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 (°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.

TAC3D (processed map)

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

CDE (PDP procedure element)

```
COMDIM* FCOPY
      PARAMETER IQ=12,JQ=12,KQ=12,NQ=15,MQ=12,IGQ=4,JGQ=4,KGQ=4,LQ=50
C=======================================================================
C     COMMON  AND  DIMENSION  FOR  RAT-3D          MARCH, 1969
C
      REAL               KR          ,KZ          ,KT
      LOGICAL            FIRST       ,GAS         ,SW          ,DP
      INTEGER            OUTTAP
C
      DIMENSION          RBL     ( LQ),RBH    ( LQ),RDG    ( LQ),MGR    ( LQ),
     -                   ZBL     ( LQ),ZBH    ( LQ),ZDG    ( LQ),MGZ    ( LQ),
     -                   TBL     ( LQ),TBH    ( LQ),TDG    ( LQ),MGT    ( LQ),
     -                   NFLO    ( NQ),CARD   (  1),
     -                   KR      ( IQ,JQ,KQ),       KZ     ( IQ,JQ,KQ),
     -                   KT      ( IQ,JQ,KQ),       MT     ( IQ,JQ,KQ)
C
      COMMON /TEST   / SW      ( 25),OUTTAP       ,DP           ,NTA
C
      COMMON /MXXX   / DTIME   ( 21),FTIME  ( 21),ITER   ( 21),ITAPE  ( 21)
C
      COMMON /IXX    / DELR    ( IQ),RP     ( IQ),RL     ( IQ),RLN    ( IQ),
     -                 IGR     ( IQ)
C
      COMMON /JXX    / DELZ    ( JQ),ZP     ( JQ),ZL     ( JQ),ZLN    ( JQ),
     -                 JGZ     ( JQ)
C
      COMMON /KXX    / DELT    ( KQ),TP     ( KQ),TL     ( KQ),TLN    ( KQ),
     -                 KGT     ( KQ)
C
      COMMON /NXX    / FLOW    ( NQ),TI     ( NQ),TO     ( NQ),IPATH  ( NQ),
     -                 IFLO    ( NQ),ITIN   ( NQ),ITI    ( NQ),ITO    ( NQ),
     -                 IF      ( NQ),FLODEP( NQ),TINDEP( NQ),
     -                 RLIM1   ( NQ),RLIM2  ( NQ),RLIM3  ( NQ),RLIM4  ( NQ),
     -                 FLIM1   ( NQ),FLIM2  ( NQ),FLIM3  ( NQ),FLIM4  ( NQ),
     -                 TLIM1   ( NQ),TLIM2  ( NQ),TLIM3  ( NQ),TLIM4  ( NQ),
     -                 RCPC    ( NQ,MQ)
C
      COMMON /IGXX   / RATIOH(IGQ),RATIOK(IGQ),RATIOC(IGQ)
C
      COMMON /JGXX   / ZATIOH(JGQ),ZATIOK(JGQ)
C
      COMMON /KGXX   / TATIOH(KGQ),TATIOK(KGQ)
C
      COMMON /LXX    / IL      ( LQ),IH     ( LQ),JL     ( LQ),JH     ( LQ),
     -                 KL      ( LQ),KH     ( LQ),MB     ( LQ)
```

```
C
      COMMON /USER   / DT          ,DR          ,FTR         ,FTZ         ,
     -                 FTT         ,GK          ,HR          ,FR          ,
     -                 RE          ,ST          ,TOUT        ,TIN
C
      COMMON /XMAX   / MAXFLO      ,MMAX        ,MAXRP       ,MAXZP       ,
     -                 MAXTP       ,MAXNB       ,MAXMAT      ,MAXRG       ,
     -                 MAXZG       ,MAXTG
C
      COMMON /XSUBS  / IM          ,IM1         ,IMAX        ,IGLS        ,
     -                 IGHS        ,NRG         ,LMAX        ,JM          ,
     -                 JM1         ,JMAX        ,JGLS        ,JGHS        ,
     -                 NZG         ,KM          ,KM1         ,KMAX        ,
     -                 KGLS        ,KGHS        ,NTG
C
      COMMON /MISCXX/  CURTI       ,FIRST       ,GAS         ,ICOUNT      ,
     -                 ISHAPE      ,NITER       ,NPRINT      ,SCALE       ,
     -                 NI          ,TB          ,TH          ,DATI        ,
     -                 NC          ,IERROR(   2),ZA      (  14),PNAME  (  12),
     -                 CS1         ,CS2         ,CS3
C
      COMMON /XMADAX/  ILS         ,IHS         ,JLS         ,JHS         ,
     -                 KLS         ,KHS
C
      COMMON /XTRA   / A1          ,A2          ,A3          ,A4          ,
     -                 A5          ,A6          ,A7          ,A8          ,
     -                 A9          ,A10         ,A11         ,A12         ,
     -                 A13         ,A14         ,A15         ,A16         ,
     -                 A17         ,A18
C
      COMMON /IGXJXK/  GAPR   (IGQ,JQ,KQ),      MATRG  (IGQ,JQ,KQ),
     -                 RBBTL  (IGQ,JQ,KQ),      RBBTH  (IGQ,JQ,KQ),
     -                 REML   (IGQ,JQ,KQ),      REMH   (IGQ,JQ,KQ),
     -                 RATIOB(IGQ,JQ,KQ)
C
      COMMON /JGXIXK/  GAPZ   (JGQ,IQ,KQ),      MATZG  (JGQ,IQ,KQ),
     -                 ZBBTL  (JGQ,IQ,KQ),      ZBBTH  (JGQ,IQ,KQ),
     -                 ZEML   (JGQ,IQ,KQ),      ZEMH   (JGQ,IQ,KQ),
     -                 ZATIOB(JGQ,IQ,KQ)
C
      COMMON /KGXIXJ/  GAPT   (KGQ,IQ,JQ),      MATTG  (KGQ,IQ,JQ),
     -                 TBBTL  (KGQ,IQ,JQ),      TBBTH  (KGQ,IQ,JQ),
     -                 TEML   (KGQ,IQ,JQ),      TEMH   (KGQ,IQ,JQ),
     -                 TATIOB(KGQ,IQ,JQ)
C
      COMMON /IXJXKX/  RR     ( IQ,JQ,KQ),      RZ     ( IQ,JQ,KQ),
```

```
     -                       RT     ( IQ,JQ,KQ),        V      ( IQ,JQ,KQ),
     -                       W      ( IQ,JQ,KQ),        CONR   ( IQ,JQ,KQ),
     -                       CONZ   ( IQ,JQ,KQ),        CONT   ( IQ,JQ,KQ),
     -                       RCP    ( IQ,JQ,KQ),        T      ( IQ,JQ,KQ),
     -                       TT    ( IQ,JQ,KQ)
C
      EQUIVALENCE            (MT     ,W       ),(RBBTL  ,RBL     ),(RBBTH  ,RBH     ),
     -                       (REML   ,RDG     ),(REMH   ,MGR     ),(ZBBTL  ,ZBL     ),
     -                       (ZBBTH  ,ZBH     ),(ZEML   ,ZDG     ),(ZEMH   ,MGZ     ),
     -                       (TBBTL  ,TBL     ),(TBBTH  ,TBH     ),(TEML   ,TDG     ),
     -                       (TEMH   ,MGT     ),(CONR   ,KR      ),(CONZ   ,KZ      ),
     -                       (CONT   ,KT      ),(CONR   ,CARD    ),(FLODEP,NFLO    )
C========================================================================
C========================================================================
      END
```

```
C======================================================================
      INCLUDE       COMDIM
C======================================================================
C     MAIN PROGRAM OF TAC3D
C======================================================================
C======================================================================
C     DEFINITIONS OF PROGRAM VARIABLES
C======================================================================
C     A1...A18            EXTRA FUNCTION VARIABLES USED IN MATERIAL
C                         AND COOLANT FUNCTIONS
C======================================================================
C     CONR(I,J,K)         INVERSE LOCAL RADIAL THERMAL CONDUCTIVITY
C                         (HR-FT-R/BTU)
C     CONT(I,J,K)         INVERSE LOCAL THETA THERMAL CONDUCTIVITY
C                         (HR-FT-R/BTU)
C     CONZ(I,J,K)         INVERSE LOCAL AXIAL THERMAL CONDUCTIVITY
C                         (HR-FT-R/BTU)
C     CURTI               CURRENT TIME,(HR)
C======================================================================
C     DATI                TIME AT WHICH THE TIMEDEPENDENT DATA SHOULD
C                         BE CALCULATED
C     DT                  CURRENT TIME INCREMENT,(HR)
C     DTIME(M)            TIME STEP USED IN TIME PERIOD M, HR.
C     DIM(2)              CONSTANT DIMENSION VALUES IN HEAT TRANSFER
C                         COEFFICIENT DETERMINATION
C     DR                  CONTAINS THE LOCAL TEMPERATURE IN DEGREES
C                         RANKINE. IT IS USED IN THE DATA FUNCTIONS.
C     DELR(I)             DISTANCE BETWEEN POINTS I AND I+1(FT)
C     DELZ(J)             DISTANCE BETWEEN POINTS J AND J+1(FT)
C     DELT(K)             DISTANCE BETWEEN POINTS K AND K+1(FT)
C======================================================================
C     FTIME(M)            TIME AT WHICH TIME PERIOD M ENDS, HR.
C     FR = FLOW(N)        LOCAL FLOW RATE, LB/HR, USED IN DATA FUNCTIONS
C     FIRST               A LOGICAL PARAMETER CONTROLLING CERTAIN ACTIONS
C                         TRUE=  SPECIAL ACTION, ONLY ONCE
C                         FALSE=  NORMAL ACTION, ALL OTHER TIMES
C     FLOW(N)             FLOW RATE OF COOLANT N, LB/HR
C                         =0, CURRENT COOLANT NOT USED
C     FLIM1(N)            LOW LIMIT, LOW RANGE FLOW RATE FUNCTION
C     FLIM2(N)            UPPER LIMIT, LOW RANGE FLOW RATE FUNCTION
C     FLIM3(N)            UPPER LIMIT, MIDDLE RANGE FLOW RATE FUNCTION
C     FLIM4(N)            UPPER LIMIT, UPPER RANGE FLOW RATE FUNCTION
C     FTR   = RP(I)       RADIAL LOCATION OF CURRENT POINT, FT
C     FTT   = TP(K)       THETA LOCATION OF CURRENT POINT, FT
C     FTZ   =ZP(J)        AXIAL LOCATION OF THE CURRENT POINT, FT
```

```
C=========================================================================
C      GAPR(IG,J,K)        RADIAL GAP INDICATOR
C                          =POSITIVE, LOCAL RADIAL GAP WIDTH IN GAPLINE IG
C                          =0.0, NO GAP OR COOLANT
C                          =-1.0E-10, COOLANT ON HIGH INDEX SIDE OF
C                                     GAPLINE IG
C                          =-2.0E-10, COOLANT ON LOW INDEX SIDE OF
C                                     GAPLINE IG
C                          =-3.0E-10, TWO COOLANTS ADJACENT AT GAPLINE IG
C      GAPZ(JG,I,K)      , AXIAL GAP INDICATOR
C                          =POSITIVE, LOCAL AXIAL GAP WIDTH IN GAPLINE JG
C                          =0.0, NO GAP OR COOLANT
C                          =-1.0E-10, COOLANT ON HIGH INDEX SIDE OF
C                                     GAPLINE JG
C                          =-2.0E-10, COOLANT ON LOW INDEX SIDE OF
C                                     GAPLINE JG
C                          =-3.0E-10, TWO COOLANTS ADJACENT AT GAPLINE JG
C      GAPT(KG,I,J)        THETA GAP INDICATOR
C                          =POSITIVE, LOCAL THETA GAP WIDTH IN GAPLINE KG
C                          =0.0, NO GAP OR COOLANT
C                          =-1.0E-10, COOLANT ON HIGH INDEX SIDE OF
C                                     GAPLINE KG
C                          =-2.0E-10, COOLANT ON LOW INDEX SIDE OF
C                                     GAPLINE KG
C                          =-3.0E-10, TWO COOLANTS ADJACENT AT GAPLINE KG
C      GAS                  LOGICAL VARIABLE , TRUE IF GAS
C      GK                   LOCAL GAS CONDUCTIVITY,(BTU/HR,FT,F)
C=========================================================================
C      HR                   CONTAINS THE TIME IN HOURS. IT IS USED IN
C                           THE DATA FUNCTIONS.
C=========================================================================
C      IGR(I)               GAP INDEX OF GAP AT LINE I
C      IM1                  IM-1
C      I                    RADIAL POINT OR LINE INDEX
C      IA                   LOCAL HEAT TRANSFER CORRELATION RANGE
C      ITER(M)              NUMBER OF TIME STEPS BETWEEN DATA EVALUATIONS
C                           IN TIME PERIOD M
C      IUNIT                CODE NUMBER INDICATING THE UNITS OF THE CURRENT
C                           TIME PERIOD DATA BEING READ.
C                           0=SAME UNITS AS BEFORE
C                           1=SEC.
C                           2=MIN.
C                           3=HR.
C      IFLO(N)              FLOW DEPENDENCE NUMBER OF COOLANT N
C                           =1, NO DEPENDENCE
C                           =2, FLOW DEPENDS ON TIME
```

```
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      MAIN1000
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                         MAIN1080
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========================================================================MAIN1430
C       K                       THETA POINT OR LINE INDEX                        MAIN1440
C       KGT(K)                  GAP INDEX OF GAP AT LINE K                       MAIN1450
C       KR(I,J,K)               AVERAGE RADIAL CONDUCTIVITY BETWEEN POINTS I ANDMAIN1460
C                               I+1                                              MAIN1470
C                               (BTU/(HR,FT,F))                                  MAIN1480
C       KZ(I,J,K)               AVERAGE AXIAL CONDUCTIVITY BETWEEN POINTS J AND MAIN1490
C                               J+1                                              MAIN1500
C                               (BTU/(HR,FT,F))                                  MAIN1510
C       KT(I,J,K)               AVERAGE THETA CONDUCTIVITY BETWEEN POINTS K AND MAIN1520
C                               K+1                                              MAIN1530
C                               (BTU/(HR,FT,F))                                  MAIN1540
C       KM1                     KM-1                                             MAIN1550
C       KG                      THETA GAP INDEX                                  MAIN1560
C       KGLS                    THE GAP NUMBER ASSOCIATED WITH THE LOW THETA     MAIN1570
C                               BOUNDARY OF THE CURRENT BLOCK                    MAIN1580
C       KGHS                    THE GAP NUMBER ASSOCIATED WITH THE HIGH THETA    MAIN1590
C                               BOUNDARY OF THE CURRENT BLOCK                    MAIN1600
C       KH(L)                   INDEX OF THE HIGH THETA POINT OF BLOCK L         MAIN1610
C       KL(L)                   INDEX OF THE LOW THETA POINT OF BLOCK L          MAIN1620
C       KHS   = KH(L)           INDEX OF THE HIGH THETA POINT OF CURRENT BLOCK   MAIN1630
C       KLS   = KL(L)           INDEX OF THE LOW THETA POINT OF CURRENT BLOCK    MAIN1640
C       KM                      NUMBER OF THETA GRIDLINES                        MAIN1650
C       KMAX                    NUMBER OF THETA POINTS                           MAIN1660
C========================================================================MAIN1670
C       LMAX                    NUMBER OF BLOCKS                                 MAIN1680
C========================================================================MAIN1690
C       MT(I,J,K)               TABLE USED TO TEST COMPLETENESS OF THE           MAIN1700
C                               BLOCK ASSIGNMENTS                                MAIN1710
C       MAXFLO                  MAXIMUM NUMBER OF COOLANTS ALLOWED               MAIN1720
C       MMAX                    MAXIMUM NUMBER OF TIME PERIODS ALLOWED           MAIN1730
C       MAXRP                   MAXIMUM NUMBER OF RADIAL POINTS ALLOWED          MAIN1740
C       MAXZP                   MAXIMUM NUMBER OF AXIAL POINTS ALLOWED           MAIN1750
C       MAXTP                   MAXIMUM NUMBER OF THETA POINTS ALLOWED           MAIN1760
C       MAXNB                   MAXIMUM NUMBER OF BLOCKS ALLOWED                 MAIN1770
C       MGR(L)                  MATERIAL NUMBER OF THE RADIAL GAP OF BLOCK L     MAIN1780
C       MGZ(L)                  MATERIAL NUMBER OF THE AXIAL GAP OF BLOCK L      MAIN1790
C       MGT(L)                  MATERIAL NUMBER OF THE THETA GAP OF BLOCK L      MAIN1800
C       MAXMAT                  MAXIMUM NUMBER OF MATERIALS AND COOLANTS ALLOWEDMAIN1810
C       MAXRG                   MAXIMUM NUMBER OF RADIAL GAPS                    MAIN1820
C       MAXZG                   MAXIMUM NUMBER OF AXIAL GAPS                     MAIN1830
```

```
C       MAXTG           MAXIMUM NUMBER OF THETA GAPS                          MAIN1840
C       M  = IPATH(N)   LOCAL FLOW DIRECTION                                  MAIN1850
C       MATRG(IG,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(JG,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(KG,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=============================================================================MAIN2160
C       N               COOLANT SUBSCRIPT                                     MAIN2170
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=============================================================================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        MAIN2340
C                                EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN2350
C       RLN(I)                   GEOMETRICAL CONSTANT USED TO CALCULATE THE        MAIN2360
C                                EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN2370
C       RATIOH(IG)               GEOMETRICAL CONSTANT USED TO CALCULATE THE        MAIN2380
C                                EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN2390
C       RATIOK(IG)               GEOMETRICAL CONSTANT USED TO CALCULATE THE        MAIN2400
C                                EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN2410
C       RBBIH(IG,J,K)            LOCAL TEMPERATURE OF THE RADIAL BOUNDARY OF A     MAIN2420
C                                BLOCK BOUNDED ON ITS HIGH I INDEX SIDE            MAIN2430
C                                BY GAP IG,(R)                                     MAIN2440
C       RBBIL(IG,J,K)            LOCAL TEMPERATURE OF THE RADIAL BOUNDARY OF A     MAIN2450
C                                BLOCK BOUNDED ON ITS LOW I INDEX SIDE             MAIN2460
C                                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 DIRECTIMAIN2490
C       RE                       LOCAL REYNOLDS NUMBER, USED IN DATA FUNCTIONS     MAIN2500
C       REMH(IG,J,K)             INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL     MAIN2510
C                                BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH I INDEX MAIN2520
C                                SIDE BY GAP IG                                    MAIN2530
C       REML(IG,J,K)             INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL     MAIN2540
C                                BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX    MAIN2550
C                                SIDE BY GAP IG                                    MAIN2560
C       RLIM1(N)                 LOW LIMIT, LOW RANGE HEAT TRANSFER CORRELATION    MAIN2570
C       RLIM2(N)                 UPPER LIMIT, LOW RANGE HEAT TRANSFER CORRELATIONMAIN2580
C       RLIM3(N)                 UPPER LIMIT, MIDDLE RANGE HEAT TRANSFER           MAIN2590
C                                CORRELATION                                       MAIN2600
C       RLIM4(N)                 UPPER LIMIT, UPPER RANGE HEAT TRANSFER            MAIN2610
C                                CORRELATION                                       MAIN2620
C       RP(I)                     RADIAL LOCATION OF POINT I,(FT)                  MAIN2630
C       RR(I,J,K)                INVERSE RADIAL RESISTANCE BETWEEN POINTS I AND IMAIN2640
C                                +1(FT)                                            MAIN2650
C       RZ(I,J,K)                INVERSE AXIAL RESISTANCE BETWEEN POINTS J AND J+MAIN2660
C                                (FT)                                              MAIN2670
C       RT(I,J,K)                INVERSE THETA RESISTANCE BETWEEN POINTS K AND K+MAIN2680
C                                1(FT)                                             MAIN2690
C=========================================================================MAIN2700
C       ST                       LOCAL SURFACE TEMPERATURE, R, USED IN DATA FUNCTMAIN2710
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     TT(I,J,K)                INTEGER VALUE OF T(I,J,K)                           MAIN2820
C     TBBIH(KG,I,J)            LOCAL TEMPERATURE OF THE THETA BOUNDARY OF A        MAIN2830
C                              BLOCK BOUNDED ON ITS HIGH K INDEX SIDE              MAIN2840
C                              BY GAP KG,(R)                                       MAIN2850
C     TBBIL(KG,I,J)            LOCAL TEMPERATURE OF THE THETA BOUNDARY OF A        MAIN2860
C                              BLOCK BOUNDED ON ITS LOW K INDEX SIDE               MAIN2870
C                              BY GAP KG,(R)                                       MAIN2880
C     TEMH(KG,I,J)             INVERSE OF THE LOCAL EMISSIVITY OF THE THETA        MAIN2890
C                              BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH K INDEX     MAIN2900
C                              SIDE BY GAP KG                                      MAIN2910
C     TEML(KG,I,J)             INVERSE OF THE LOCAL EMISSIVITY OF THE THETA        MAIN2920
C                              BOUNDARY OF A BLOCK BOUNDED ON ITS LOW K INDEX      MAIN2930
C                              SIDE BY GAP KG                                      MAIN2940
C     TLIM1(N)                 LOW LIMIT, LOW RANGE INLET TEMPERATURE FUNCTION     MAIN2950
C     TLIM2(N)                 HIGH LIMIT, LOW RANGE INLET TEMPERATURE FUNCTIONMAIN2960
C     TLIM3(N)                 HIGH LIMIT, MIDDLE RANGE INLET TEMPERATURE          MAIN2970
C                              FUNCTION                                            MAIN2980
C     TLIM4(N)                 HIGH LIMIT, UPPER RANGE INLET TEMPERATURE           MAIN2990
C                              FUNCTION                                            MAIN3000
C     TOUT  = TO(N)            LOCAL COOLANT OUTLET TEMPERATURE, R, USED IN        MAIN3010
C                              DATA FUNCTIONS.                                     MAIN3020
C     TIN  = TI(N)             LOCAL COOLANT INLET TEMPERATURE, R, USED IN DATAMAIN3030
C                              FUNCTIONS                                           MAIN3040
C     TO(N)                    COOLANT OUTLET TEMPERATURE                          MAIN3050
C     TI(N)                    COOLANT INLET TEMPERATURE                           MAIN3060
C     TH                       LOCAL WALL TEMPERATURE,(R).                         MAIN3070
C     TL                       LOCAL WALL TEMPERATURE,(R).                         MAIN3080
C     TP(K)                    THETA LOCATION OF POINT K, (FT)                     MAIN3090
C     TAT1OK(KG)               GEOMETRICAL CONSTANT USED TO CALCULATE THE          MAIN3100
C                              EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3110
C     TAT1OH(KG)               GEOMETRICAL CONSTANT USED TO CALCULATE THE          MAIN3120
C                              EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3130
C     TLN(K)                   GEOMETRICAL CONSTANT USED TO CALCULATE THE          MAIN3140
C                              EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3150
C     TAT1OB(KG,I,J)             GEOMETRICAL CONSTANT USED TO CALCULATE THE        MAIN3160
C                              EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3170
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
```

```
C=======================================================================
C       X1                  CURRENT TIME (MINUTES)
C       X2                  CURRENT TIME (SECONDS)
C=======================================================================
C       ZBL(L)              LOW AXIAL BOUNDARY OF BLOCK L (FT)
C       ZL(J)               AXIAL GRIDLINE (FT)
C       ZBH(L)              HIGH AXIAL BOUNDARY OF BLOCK L (FT)
C       ZDG(L)              AXIAL GAP THICKNESS (FT)
C       ZBBTH(JG,I,K)       LOCAL TEMPERATURE OF THE AXIAL BOUNDARY OF A
C                           BLOCK BOUNDED ON ITS HIGH J INDEX SIDE
C                           BY GAP JG,(R)
C       ZBBTL(JG,I,K)       LOCAL TEMPERATURE OF THE AXIAL BOUNDARY OF A
C                           BLOCK BOUNDED ON ITS LOW J INDEX SIDE
C                           BY GAP JG,(R)
C       ZEMH(JG,I,K)        INVERSE OF THE LOCAL EMISSIVITY OF THE AXIAL
C                           BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH J INDEX
C                           SIDE BY GAP JG
C       ZEML(JG,I,K)        INVERSE OF THE LOCAL EMISSIVITY OF THE AXIAL
C                           BOUNDARY OF A BLOCK BOUNDED ON ITS LOW J INDEX
C                           SIDE BY GAP JG
C       ZP(J)               AXIAL LOCATION OF POINT J,(FT)
C       ZAT1OK(JG)          GEOMETRICAL CONSTANT USED TO CALCULATE THE
C                           EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY
C       ZAT1OH(JG)          GEOMETRICAL CONSTANT USED TO CALCULATE THE
C                           EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY
C       ZAT1OB(JG,I,K)      GEOMETRICAL CONSTANT USED TO CALCULATE THE
C                           EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY
C       ZLN(J)              GEOMETRICAL CONSTANT USED TO CALCULATE THE
C                           EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY
C=======================================================================
C=======================================================================
C=======================================================================
C       READ AND PROCESS THE INPUT DATA OF THE NEXT PROBLEM
   20 CALL MP1
C       HAS ANY ERROR BEEN FOUND? -'YES,NO'-
        IF (IERROR(1).NE.0) GO TO 20
C       CALCULATE AND PRINT THE TEMPERATURE DISTRIBUTIONS
        CALL MP2
        GO TO 20
C=======================================================================
        END
```

```
      SUBROUTINE MP1
      INCLUDE      COMDIM
C=======================================================================
CB    READ AND PROCESS THE INPUT DATA OF ONE PROBLEM
C=======================================================================
C=======================================================================
C     ASSIGN THE FIXED CONSTANTS
      OUTTAP=15
      MAXRP=IQ
      MAXZP=JQ
      MAXTP=KQ
      MAXRG=IGQ
      MAXZG=JGQ
      MAXTG=KGQ
      MAXMAT=15
      MAXFLO=NQ
      MAXNB=LQ
      NMAX=20
      WRITE (6,100) MAXRP,MAXZP,MAXTP,MAXRG,MAXZG,MAXTG,MAXNB
  100 FORMAT(1H1,9X,19HT A C 3 D   C O D E,5X,16HSTANDARD VERSION
     2 ,/,30X,5HIQ = ,I2,4X,5HJQ = ,I2,4X,5HKQ = ,I2,4X,6HIGQ = ,I2,4X,
     36HJGQ = ,I2,4X,6HKGQ = ,I2,4X,5H LQ = ,I3////)
C=    =================================================================
      IERROR(1)= 0
      IERROR(2)= 0
      ICOUNT=0
      DO 10 I=1,25
      SW(I)=.FALSE.
   10 CONTINUE
C=======================================================================
C     READ AND PRINT THE TITLE AND GEOMETRICAL DATA
      CALL INPUT
C     CHECK THE GEOMETRICAL DATA
      CALL CHECK
C     ASSIGN THE POINTS, BLOCKS AND GAPS
      CALL POINTS
C     CALCULATE THE CONSTANTS USED TO CORRECT FOR GAPS AND COOLANTS .
      CALL CONSTA
C     CALCULATE THE GEOMETRICAL PART OF THE RESISTANCES BETWEEN POINTS
      CALL GEOMET
C     PRINT THE LOCATION OF POINTS AND BOUNDARIES
      CALL DPRINT
C     READ THE INITIAL TEMPERATURE DISTRIBUTION
      CALL INITEM
C     READ AND PRINT THE COOLANT DATA
```

```
      CALL FLOWCA
C     READ AND PRINT THE TIMESTEP SEQUENCE
      CALL TIME
C     READ AND PRINT THE GROUP OF SPECIAL FUNCTION PARAMETERS
      CALL EXTRA
C     READ THE PREVIOUS TEMPERATURE DISTRIBUTION, IF PRESENT
      CALL PRETEM
      CALL OTCARD
C     STOP EXECUTION IF ANY ERRORS HAVE BEEN FOUND
      IF(IERROR(1).NE.0) STOP
C=    ==================================================================
C     INITIALIZE ARRAYS
      DO 20 IG=1,IGQ
      DO 20 J=1,JQ
      DO 20 K=1,KQ
      REML(IG,J,K) = 0.0
      REMH(IG,J,K) = 0.0
   20 CONTINUE
      DO 30 JG=1,JGQ
      DO 30 I=1,IQ
      DO 30 K=1,KQ
      ZEML(JG,I,K)=0.0
      ZEMH(JG,I,K)=0.0
   30 CONTINUE
      DO 40 KG=1,KGQ
      DO 40 I=1,IQ
      DO 40 J=1,JQ
      TEML(KG,I,J)=0.0
      TEMH(KG,I,J)=0.0
   40 CONTINUE
C=======================================================================
C     SHOULD THE OUTPUT BE PUT ON A TAPE? -'NO,YES'-
      IF(.NOT.SW(3))RETURN
C     PUT THE INITIAL RECORD (TITLE,GRID INFORMATION) ON THE TAPE
      ILEN=1
      WRITE (OUTTAP) ISHAPE,ILEN,IMAX,JMAX,KMAX,LMAX,MAXFLO,NRG,NZG,NTG,
     1(ZA(I),I=1,12),(RL(I),I=1,IMAX),(ZL(J),J=1,JMAX),(TL(K),K=1,KMAX),
     -(IGR(I),I=1,IMAX),(JGZ(J),J=1,JMAX),(KGT(K),K=1,KMAX),
     -(IL(L),IH(L),JL(L),JH(L),KL(L),KH(L),MB(L),L=1,LMAX)
C=======================================================================
      RETURN
C*    ******************************************************************
C*    ******************************************************************
C*    ******************************************************************
      SUBROUTINE OTCARD
      DIMENSION       FORMT (  8),FORM  (  8),FB    (  2)
```

```
      DATA FORMT      /6H( 10X,, 6*6H  12X,, 6HA6,A2)/
      DATA FB         /6HF12.5,, 6HE12.5,/
      WRITE (6,90)(ZA(I),I=1,12)
   90 FORMAT(1H1,30X,12A6,///,40X,21HPROBLEM  INPUT  CARDS,//,10X,     80H
     1    0    1    1    2    2    3    3    4    4    5    5    6    6
     2    7    7    8,/,10X,                                         80H
     3    5    0    5    0    5    0    5    0    5    0    5    0    5
     4    0    5    0,/,10X,                                         80H
     5XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     6XXXXXXXXXXXXXX,//)
      IC=0
  100 IC=IC+1
      IF (IC.GT.NC) RETURN
      ITYP=CARD(IC)+0.1
      IF (ITYP.EQ.0) GO TO 150
      IF (ITYP.EQ.7) GO TO 160
      DO 120 I=1,8
      FORM(I)=FORMT(I)
  120 CONTINUE
      DO 130 I=1,ITYP
      FORM(I+1)=FB(1)
      J=IC+I
      IF (CARD(J).NE.0.0) GO TO 125
      CARD(J)=0.0
      GO TO 130
  125 IF(CARD(J) .LT. 0.0001 .OR. CARD(J) .GT. 9999.9999)FORM(I+1)=FB(2)
  130 CONTINUE
  140 I1=IC+1
      I2=IC+ITYP+2
      WRITE (6,FORM) (CARD(I),I=I1,I2)
      GO TO 180
  150 WRITE (6,155)
  155 FORMAT(82X,5HBLANK)
      GO TO 100
  160 I1=IC+1
      I2=IC+14
      WRITE (6,170) (CARD(I),I=I1,I2)
  170 FORMAT (10X,13A6,A2)
  180 IC=I2
      GO TO 100
      RETURN
C*    *****************************************************************
C*    *****************************************************************
      END
```

```
      SUBROUTINE INPUT
      INCLUDE       COMDIM
C==================================================================
C     READ THE GEOMETRY DATA.
C==================================================================
C     ERROR STOPS=
C         INPUT1     THE GEOMETRY TYPE DESIRED HAS BEEN MISSPELLED.
C         INPUT2     THE LOW RADIAL-X BLOCK BOUNDARY OF SOME BLOCK DOES
C                    NOT COINCIDE WITH A RADIAL-X GRID LINE
C         INPUT3     THE HIGH RADIAL-X BLOCK BOUNDARY OF SOME BLOCK DOES
C                    NOT COINCIDE WITH A RADIAL-X GRID LINE
C         INPUT4     THE LOW AXIAL-Z BLOCK BOUNDARY OF SOME BLOCK
C                    DOES NOT COINCIDE WITH A AXIAL-Z GRID LINE
C         INPUT5     THE HIGH AXIAL-Z BLOCK BOUNDARY OF SOME BLOCK
C                    DOES NOT COINCIDE WITH AN AXIAL-Z GRID LINE
C         INPUT6     THE LOW THETA-Y BLOCK BOUNDARY OF SOME BLOCK
C                    DOES NOT COINCIDE WITH A THETA-Y GRID LINE
C         INPUT7     THE HIGH THETA-Y BLOCK BOUNDARY OF SOME BLOCK
C                    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    Z/
      DATA BLANKS       /6H       /
      DATA BLANK        /6H       /
      DATA STAR         /5H*     /
      DATA GAP          /5H#     /
      DATA DOT          /5H.     /
C==================================================================
      PNAME(4) = TYPE(6)
```

```
      PNAME(5) = TYPE(7)
      PNAME(6) = TYPE(6)
      PNAME(7) = TYPE(7)
      NC=0
C======================================================================
C     READ AND PRINT THE TITLE
      CALL TITLE
C======================================================================
C     READ THE NEXT OPTION
    1 READ (5,103) (OPTION(I),I=1,14)
  103 FORMAT(13A6,A2)
      WRITE (6,100) (OPTION(I), I= 1,12)
  100 FORMAT (1H0,19X,12A6)
C     HAVE ALL OPTIONS BEEN READ? -'YES,NO'-
      IF(OPTION(2).EQ.BLANKS)GO TO 4
      CALL ACARD(OPTION)
C/    ASSIGN THE OPTION -'OK,ERROR'-
      DO 2 I=1,20
      IF(OPTION(2).NE.OPT(I))GO TO 2
      SW(I)=.TRUE.
      GO TO 1
    2 CONTINUE
CE    *****************************************************************
C     OPTION COULD NOT BE FOUND, PRINT A WARNING
      WRITE(6,99)OPTION(2)
   99 FORMAT(8H1OPTION ,A6,18H COULD NOT BE READ)
      GO TO 1
C======================================================================
    4 NC=NC+1
      CARD(NC)=0.0
C     SPECIFY THE PRINT FORMAT
      NPRINT=3
      IF(SW(17)) NPRINT=1
      IF(SW(18))NPRINT=2
      IF(SW(19)) NPRINT=3
C=    =================================================================
C     IS IT A RECTANGULAR GEOMETRY? -'YES,NO'-
      IF(SW(14))GO TO 5
C     IS IT A CYLINDRICAL GEOMETRY? -'YES,NO'-
      IF(SW(15))GO TO 3
C     THE TYPE OF GEOMETRY IS NOT SPECIFIED CORRECTLY
      CALL ERROR(6HINPUT1)
C======================================================================
C     SET THE PROBLEM UP FOR A CYLINDRICAL GEOMETRY
    3 SCALE=57.2957795
      ISHAPE=0
```

```
      PNAME(1) = TYPE(1)
      PNAME(2) = TYPE(2)
      PNAME(3)=TYPE(3)
      PNAME(4)=TYPE(8)
      PNAME(5)=TYPE(9)
      GO TO 10
C=====================================================================
C     SET THE PROBLEM UP FOR A RECTANGULAR GEOMETRY
    5 SCALE=12.0
      ISHAPE=1
      PNAME(1) = TYPE(4)
      PNAME(2) = TYPE(5)
      PNAME(3)=TYPE(10)
C     ================================================================
CB    READ AND PROCESS THE GRID DESCRIPTIONS
   10 IM=MAXRP-1
C     READ THE INPUT DESCRIBING THE RADIAL-X GRID LINES
      CALL GRID (RL,IM)
      JM= MAXZP-1
C     READ THE INPUT DESCRIBING THE AXIAL-Z GRID LINES
      CALL GRID (ZL,JM)
      KM= MAXTP-1
C     READ THE INPUT DESCRIBING THE THETA-Y GRID LINES
      CALL GRID (TL,KM)
C=====================================================================
C     ASSIGN THE GRID LIMITS
      IMAX=IM+1
      JMAX=JM+1
      KMAX=KM+1
      IM1=IM-1
      JM1=JM-1
      KM1=KM-1
C=====================================================================
C     CHANGE A ZERO INNER RADIUS TO A SMALL BUT FINITE ONE
      IF (RL(1) .LE. 0.0)  RL(1) = 1.0E-5
C     CHANGE THE RADIAL-X GRID LINE DATA TO THE CORRECT UNITS
      DO 20 I=1,IM
      RL(I)=RL(I)/12.0
   20 CONTINUE
C=====================================================================
C     CHANGE THE AXIAL-Z GRID LINE DATA TO THE CORRECT UNITS
      DO 30 J=1,JM
      ZL(J)=ZL(J)/12.0
   30 CONTINUE
C=====================================================================
C     CHANGE THE THETA-Y GRID LINE DATA TO THE CORRECT UNITS
```

```
      DO 40 K=1,KM
      TL(K)=TL(K)/SCALE
   40 CONTINUE
C=======================================================================
C     PRINT THE HEADING OF THE BLOCK PRINT
      WRITE (6,105) (ZA(I), I= 1,12), PNAME(1), PNAME(1), PNAME(2),
     -PNAME(2), PNAME(3), PNAME(3), PNAME(1), PNAME(2), PNAME(3),
     -PNAME(4), PNAME(5), PNAME(4), PNAME(5), PNAME(4), PNAME(5)
  105 FORMAT (1H1,30X,12A6,///,50X,18HBLOCK  DESCRIPTION,//,2X,5HBLOCK,
     -3X,66H.   .   . .   .   .   .   .   .   BOUNDARIES   .   .   .   .   .   .
     - .   . ,9X,44H.   .   .   .   .   GAPS   .   .   .   .   .   ./1X,
     -6HNUMBER/8X,4HLOW ,A6,6H HIGH ,A6,5H LOW ,A6,5HHIGH ,A6,4HLOW ,
     -A6,5HHIGH ,A6,8HMATERIAL,4X,A6,11H MATERIAL  ,A6,10HMATERIAL  ,A6,
     -8HMATERIAL/9X,8H(INCHES),3X,8H(INCHES),4X,8H(INCHES),2X,8H(INCHES)
     -,3X,A6,A3,1X,A6,A3,13X,8H(INCHES),8X,8H(INCHES),8X,A6,A3//)
C=======================================================================
CB    READ AND PRINT THE BLOCK DESCRIPTIONS
      K=0
C     COUNT THE BLOCKS
   50 K=K+1
C     READ THE MATERIAL NUMBER
      READ (5,101) X,Y,FLAG1,FLAG2
  101 FORMAT(2E12.4,48X,A6,A2)
C     HAVE ALL THE BLOCK CARDS BEEN READ? -'YES,NO'-
      IF(X.EQ.0.0) GO TO 60
      CALL FCARD(2.0,X,Y,FLAG1,FLAG2,DUM,DUM,DUM,DUM)
C     IS IT A COOLANT? -'NO,YES'-
      IF (X.GT.0.0) GO TO 55
C     ASSIGN THE FLOW DIRECTION OF THE COOLANT
      N=-(X-0.1)
      IPATH(N)= Y
C     ASSIGN THE COOLANT OR MATERIAL NUMBER
   55 MB(K)=X
C=======================================================================
C     READ THE BLOCK DIMENSIONS
      READ (5,104) RBL(K),RBH(K),ZBL(K),ZBH(K),TBL(K),TBH(K),FLAG1,FLAG2
  104 FORMAT(6E12.4,A6,A2)
      CALL FCARD(6.0,RBL(K),RBH(K),ZBL(K),ZBH(K),TBL(K),TBH(K),FLAG1,
     XFLAG2)
C=======================================================================
C     READ THE GAP PROPERTIES
      READ (5,104) RDG(K),X,ZDG(K),Y,TDG(K),Z,FLAG1,FLAG2
      CALL FCARD(6.0,RDG(K),X,ZDG(K),Y,TDG(K),Z,FLAG1,FLAG2)
      MGR(K)=X
      MGZ(K)=Y
      MGT(K)=Z
```

```
C=============================================================================
C      PRINT THE BLOCK DEFINITION
       IND=0
       DO 240 I=1,11
       FORMT(I)=FORM(I)
  240  CONTINUE
       DO 308 I=1,3
       GO TO (301,302,303),I
  301  IF (X.LE.0.0) GO TO 308
       MG=MGR(K)
       GG=RGG(K)
       GO TO 305
  302  IF (Y.LE. 0.0) GO TO 308
       MG=MGZ(K)
       GG=ZGG(K)
       GO TO 305
  303  IF (Z .LE. 0.0) GO TO 308
       MG=MGT(K)
       GG=TGG(K)
  305  IND=IND+1
       MATGAP(IND)=MG
       DIMGAP(IND)=GG
       I1=I+2
       FORMT(I1+3)=GPRINT(1)
       FORMT(I1+4)=GPRINT(2)
  308  CONTINUE
       IF (IND.EQ. 0) GO TO 311
       WRITE (6,FORMT)K,RBL(K),RBH(K),ZBL(K),ZBH(K),TBL(K),TBH(K),MB(K),
      1(DIMGAP(I),MATGAP(I),I=1,IND)
       GO TO 50
  311  WRITE (6,FORMT)K,RBL(K),RBH(K),ZBL(K),ZBH(K),TBL(K),TBH(K),MB(K)
       GO TO 50
C=============================================================================
C      ASSIGN THE NUMBER OF BLOCKS
   60  NC=NC+1
       CARD(NC)=0.0
       LMAX=K-1
C/     CONVERT THE BLOCK AND GAP DIMENSIONS TO THEIR CORRECT UNITS
       DO 70 K=1,LMAX
       IF (RBL(K) .LE. 0.0)  RBL(K) = 1.0E-5
       RBL(K)=RBL(K)/12.0
       IF (RBH(K) .LE. 0.0)  RBH(K) = 1.0E-5
       RBH(K)=RBH(K)/12.0
       ZBL(K)=ZBL(K)/12.0
       ZBH(K)=ZBH(K)/12.0
       TBL(K)=TBL(K)/SCALE
```

```
      TBH(K)=TBH(K)/SCALE
      RDG(K)=RDG(K)/12.0
      ZDG(K)=ZDG(K)/12.0
      TDG(K)=TDG(K)/SCALE
   70 CONTINUE
CE    **********************************************************************
C     TEST THAT BLOCK BOUNDARIES COINCIDE WITH GRID LINES
      DO 261 I=1,IMAX
      DO 261 J=1,JMAX
      DO 261 K=1,KMAX,
      BTABLE(I,J,K)= BLANK
  261 CONTINUE
      DO 400 L=1,LMAX
      DO 270 ILG=1,IM
      IF (RBL(L).GE.0.99999*RL(ILG).AND.RBL(L).LE.1.00001*RL(ILG)) GO TO
     1 280
  270 CONTINUE
      CALL ERROR (6HINPUT2)
  280 DO 290 IHG=1,IM
      IF (RBH(L).GE.0.99999*RL(IHG).AND.RBH(L).LE.1.00001*RL(IHG)) GO TO
     1 300
  290 CONTINUE
      CALL ERROR (6HINPUT3)
  300 DO 310 JLG=1,JM
      IF (ZBL(L).GE.0.99999*ZL(JLG).AND.ZBL(L).LE.1.00001*ZL(JLG)) GO TO
     1 320
  310 CONTINUE
      CALL ERROR (6HINPUT4)
  320 DO 330 JHG=1,JM
      IF (ZBH(L).GE.0.99999*ZL(JHG).AND.ZBH(L).LE.1.00001*ZL(JHG)) GO TO
     1 340
  330 CONTINUE
      CALL ERROR (6HINPUT5)
  340 DO 350 KLG=1,KM
      IF (TBL(L) .GE. 0.99999*TL(KLG) .AND. TBL(L) .LE. 1.00001*TL(KLG))
     XGO TO 360
  350 CONTINUE
      CALL ERROR (6HINPUT6)
  360 DO 370 KHG=1,KM
      IF (TBH(L) .GE. 0.99999*TL(KHG) .AND. TBH(L) .LE. 1.00001*TL(KHG))
     XGO TO 331
  370 CONTINUE
      CALL ERROR (6HINPUT7)
  331 IF (MB(L).GT.0) GO TO 335
      DO 332 J=JLG,JHG
      DO 332 K=KLG,KHG
```

```
      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*  WHER
     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    ******************************************************************
```

```
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(RL1,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=========================================================================
      ILS= 1
      IHS= 6
C     READ THE NEXT GRID DATA CARD
  100 READ (5,140) (RL(I),I=1,8)
C     HAVE ALL GRID DATA CARDS BEEN READ? -'YES,NO'-
      IF (RL(1).EQ.0.0.AND.RL(2).EQ.0.0) GO TO 120
      CALL FCARD(6,0,RL(1),RL(2),RL(3),RL(4),RL(5),RL(6),RL(7),RL(8))
C=========================================================================
      IM= MIN0(IHS,IM1)
C     ARE THERE TOO MANY GRID DATA CARDS ? -'YES,NO'-
C     PRINT ERROR GRID 1
      IF (ILS.GT.IM) CALL ERROR(6H GRID1)
C     RECORD THE CONTENTS OF THE CURRENT GRID DATA CARD
      DO 110 I= ILS,IM
      J= I+1-ILS
      RL1(I)= RL(J)
  110 CONTINUE
C     PREPARE FOR THE READING OF THE NEXT GRID DATA CARD
      ILS= IHS+1
      IHS= IHS+6
      GO TO 100
C=========================================================================
C/    DETERMINE THE NUMBER OF GRIDLINES READ
  120 NC=NC+1
      CARD(NC)=0.00
      IM=ILS
  130 IM=IM-1
      IF (RL1(IM).LE.0.0) GO TO 130
      IM1= IM
CE    *******************************************************************
      RETURN
C=========================================================================
  140 FORMAT (6E12.4,A6,A2)
C=========================================================================
CE    *******************************************************************
CE    *******************************************************************
CE    *******************************************************************
      SUBROUTINE BARRAY
C=    ===================================================================
      DIMENSION        FORMN1(  3),FORMN2(  3),FORMA1(  3),FORMA2(  3),
     1                 FORMN (  3),FORMA (  3)
      DIMENSION        NUMBER( 50)
      DATA (FORMN1(I),I=1,3)   /5H(1H0,, 3H6X,, 5H25I5)/
      DATA (FORMN2(I),I=1,3)   /5H(1H0,, 3H1X,, 5H26I5)/
      DATA (FORMA1(I),I=1,3)   /4H(I4,, 3H7X,, 5H24A5)/
```

```
      DATA (FORMA2(I),I=1,3)    /4H(I4,, 3H2X,, 5H25A5)/
      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/
C=======================================================================
C     WRITE THE PROPER HEADING
      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=JM
      N=KM
      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=JM
      N=IM
      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=KM
      N=IM
   40 DO 600 M=1,MM
      NUM1=1
      NUM2=24
      IF (NUM2 .GE. N) NUM2=N
      DO210 I=1,3
      FORMN(I)=FORMN1(I)
      FORMA(I)=FORMA1(I)
  210 CONTINUE
      WRITE (6,4) PNAME(NPRINT),M
    4 FORMAT(1H0,/,48X,A6,2H (,I2,7H) PLANE)
  150 WRITE (6,FORMN) (NUMBER(L),L=NUM1,NUM2)
      WRITE (6,170)
  170 FORMAT (1H0)
      GO TO (200,300,400),NPRINT
C=======================================================================
C     OUTPUT PRINT IN RADIAL-X DIRECTION, THETA-Y DIRECTION HORIZONTAL,
C     AXIAL-Z DIRECTION VERTICAL
C=======================================================================
  200 DO 420 J=1,JM
  420 WRITE (6,FORMA) J,(BTABLE(M,J,K) ,K=NUM1,NUM2)
      IF(NUM2 .LT. N) GO TO 100
      GO TO 600
```

```
C=======================================================================
C     OUTPUT PRINT IN AXIAL-Z DIRECTION, RADIAL-X DIRECTION HORIZONTAL,
C     THETA-Y DIRECTION VERTICAL
C=======================================================================
  300 DO 320 K=1,KM
  320 WRITE (6,FORMA)  K,(BTABLE(I,M,K) , I=NUM1,NUM2)
      IF(NUM2 .LT. N) GO TO 100
      GO TO 600
C=======================================================================
C     OUTPUT PRINT IN,THETA-Y DIRECTION, RADIAL-X DIRECTION HORIZONTAL,
C     AXIAL-Z DIRECTION VERTICAL
C=======================================================================
  400 DO 230 J=1,JM
  230 WRITE (6,FORMA) J, (BTABLE(I,J,M),I=NUM1,NUM2)
      IF(NUM2 .LT. N) GO TO 100
      GO TO 600
  100 NUM1=NUM2+1
      NUM2=NUM2 + 25
      IF (NUM2 .GE. N) NUM2=N
      DO 110 I=1,3
      FORMN(I)=FORMN2(I)
      FORMA(I)=FORMA2(I)
  110 CONTINUE
      GO TO 150
  600 CONTINUE
      RETURN
C=======================================================================
      SUBROUTINE ACARD(AC)
C=    ==================================================================
      DIMENSION        AC       ( 14)
C=    ==================================================================
      NC=NC+1
      CARD(NC)=7.0
      DO 100 I=1,14
      NC=NC+1
      CARD(NC)=AC(I)
  100 CONTINUE
      RETURN
CE    ******************************************************************
CE    ******************************************************************
      END
```

```
      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=========================================================================
      IERROR(1) = IB(1)
      IERROR(2) = IB(2)
      IF((IERROR(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
```

```
      SUBROUTINE FCARD (ANUM,C1,C2,C3,C4,C5,C6,C7,C8)
      INCLUDE       COMDIM
C=    ==================================================================
C=    ==================================================================
      DIMENSION        C      (  8)
C=    ==================================================================
      C(1)=C1
      C(2)=C2
      C(3)=C3
      C(4)=C4
      C(5)=C5
      C(6)=C6
      C(7)=C7
      C(8)=C8
      NC=NC+1
      CARD(NC)=ANUM
      II=ANUM+0.1
      II=II+2
      DO 100 I=1,II
      NC=NC+1
      CARD(NC)=C(I)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CHECK
      INCLUDE      COMDIM
C=====================================================================
C     CHECK THE GEOMETRY DATA
C=====================================================================
C     ERROR STOPS=
C        CHECK1     THERE ARE TOO MANY RADIAL-X POINTS AND GRID LINES
C        CHECK2     THERE ARE TOO MANY AXIAL-Z POINTS AND GRID LINES
C        CHECK3     THERE ARE TOO MANY THETA-Y POINTS AND GRID LINES
C        CHECK4     THE RADIAL-X GRID DATA IS OUT OF ORDER
C        CHECK5     THE AXIAL-Z GRID DATA IS OUT OF ORDER
C        CHECK6     THE THETA-Y GRID DATA IS OUT OF ORDER
C        CHECK7     THERE ARE TOO MANY BLOCKS.
C        CHECK8     THE LOW RADIAL-X BOUNDARY IS LARGER THAN THE HIGH
C                   RADIAL-X BOUNDARY FOR SOME BLOCK
C        CHECK9     THE LOW AXIAL-Z BOUNDARY IS GREATER THAN THE
C                   HIGH AXIAL-Z BOUNDARY FOR SOME BLOCK
C        CHECK10    THE LOW THETA-Y BOUNDARY IS GREATER THAN THE
C                   HIGH THETA-Y BOUNDARY FOR SOME BLOCK
C        CHECK11    THE MATERIAL NUMBER FOR A BLOCK IS LARGER THAN THE
C                   MAXIMUM NUMBER OF MATERIALS AND COOLANTS ALLOWED.
C        CHECK12    A RADIAL-X GAP MATERIAL NUMBER IS TOO HIGH
C        CHECK13    AN AXIAL-Z GAP MATERIAL NUMBER IS TOO HIGH
C        CHECK14    AN THETA-Y GAP MATERIAL NUMBER IS TOO HIGH
C     ===============================================================
C=====================================================================
C     ARE THERE TOO MANY RADIAL-X GRIDLINES? -'YES,NO'-
C     PRINT ERROR CHECK1
      IF (IMAX.GT.MAXRP) CALL ERROR(6HCHECK1)
C     ARE THERE TOO MANY AXIAL-Z GRIDLINES? -'YES,NO'-
C     PRINT ERROR CHECK2
      IF (JMAX.GT.MAXZP) CALL ERROR(6HCHECK2)
C     ARE THERE TOO MANY THETA-Y GRIDLINES? -'YES,NO'-
C     PRINT ERROR CHECK3
      IF (KMAX.GT.MAXTP) CALL ERROR(6HCHECK3)
C=====================================================================
C/    CHECK TO SEE THAT THE RADIAL-X GRID LINES ARE IN AN ASCENDING .
C    -ORDER
      DO 10 I=2,IM
      IF(RL(I).LE.RL(I-1)) CALL ERROR(6HCHECK4)
   10 CONTINUE
CE    ***************************************************************
C/    CHECK TO SEE THAT THE AXIAL-Z GRIDLINES ARE IN AN ASCENDING ORDER
      DO 20 J=2,JM
      IF (ZL(J).LE.ZL(J-1)) CALL ERROR(6HCHECK5)
```

```
   20 CONTINUE
CE    *******************************************************************
C/    CHECK TO SEE THAT THE THETA-Y GRIDLINES ARE IN AN ASCENDING ORDER
      DO 30 K=2,KM
      IF (TL(K).LE.TL(K-1)) CALL ERROR(6HCHECK6)
   30 CONTINUE
CE    *******************************************************************
C     ARE THERE TOO MANY BLOCKS? -'YES,NO'-
C     PRINT ERROR CHECK7
      IF (LMAX.GT.MAXNB) CALL ERROR (6HCHECK7)
C=========================================================================
CB    CHECK THE BLOCK GEOMETRY FOR INTERNAL CONSISTENCY
C     CHECK EVERY BLOCK IN TURN
      DO 40 L=1,LMAX
C     ARE THE RADIAL-X BOUNDARIES IN ORDER? -'NO,YES'-
C     PRINT ERROR CHECK8
      IF (RBL(L).GT.RBH(L)) CALL ERROR(6HCHECK8)
C     ARE THE AXIAL-Z BOUNDARIES IN ORDER? -'NO,YES'-
C     PRINT ERROR CHECK9
      IF (ZBL(L).GT.ZBH(L)) CALL ERROR(6HCHECK9)
C     ARE THE THETA-Y BOUNDARIES IN ORDER? -'NO,YES'-
C     PRINT ERROR CHECK10
      IF (TBL(L).GT.TBH(L)) CALL ERROR(7HCHECK10)
C     IS THE MATERIAL NUMBER A VALID ONE? -'NO,YES'-
C     PRINT ERROR CHECK11
      IF (IABS(MB(L)).GT.MAXMAT) CALL ERROR(7HCHECK11)
C     IS THE MATERIAL NUMBER OF THE RADIAL-X GAP A VALID ONE? -'NO,YES'-
C     PRINT ERROR CHECK12
      IF (MGR(L).GT.MAXMAT) CALL ERROR(7HCHECK12)
C     IS THE MATERIAL NUMBER OF THE AXIAL-Z GAP A VALID ONE? -'NO,YES'-
C     PRINT ERROR CHECK13
      IF (MGZ(L).GT.MAXMAT) CALL ERROR(7HCHECK13)
C     IS THE MATERIAL NUMBER OF THE THETA-Y GAP A VALID ONE? -'NO,YES'-
C     PRINT ERROR CHECK14
      IF (MGT(L).GT.MAXMAT) CALL ERROR(7HCHECK14)
C     HAVE ALL BLOCKS BEEN CHECKED ? -'YES,NO'-
   40 CONTINUE
CE    *******************************************************************
      RETURN
C=========================================================================
      END
```

```
      SUBROUTINE POINTS
      INCLUDE      COMDIM
C     ==================================================================
C     ASSIGN THE POINTS, BLOCKS AND GAPS
C     ==================================================================
C     ERROR STOPS=
C        POINTS1    A PART OF THE  SYSTEM WAS NOT DESCRIBED BY ANY BLOCK.
C        POINTS2    THERE ARE TOO MANY RADIAL-X GAPS.
C        POINTS3    THERE ARE TOO MANY AXIAL-Z GAPS.
C        POINTS4    THERE ARE TOO MANY THETA-Y GAPS.
C        POINTS5    A PART OF THE SYSTEM HAS BEEN DESCRIBED BY MORE THAN
C                   ONE BLOCK.
C        POINTS6    NO FLOW DIRECTION HAS BEEN ASSIGNED FOR SOME COOLANT.
C        POINTS7    1. AN EXTERNAL COOLANT IS FLOWING INTO A RADIAL-X
C                      BOUNDARY,OR
C                   2. AN INTERNAL RADIAL-X FLOW COOLANT BLOCK IS NOT
C                      TRAVERSED BY AT LEAST ONE RADIAL-X GRID LINE.
C        POINTS8    1. AN EXTERNAL COOLANT IS FLOWING INTO AN AXIAL-Z
C                      BOUNDARY,OR
C                   2. AN INTERNAL AXIAL-Z FLOW COOLANT BLOCK IS NOT
C                      TRAVERSED BY AT LEAST ONE AXIAL-Z GRID LINE.
C        POINTS9    1. AN EXTERNAL COOLANT IS FLOWING INTO A THETA-Y
C                      BOUNDARY,OR
C                   2. AN INTERNAL THETA-Y FLOW COOLANT BLOCK IS NOT
C                      TRAVERSED BY AT LEAST ONE THETA-Y GRID LINE.
C        POINTS10   A GAP HAS BEEN SPECIFIED ON THE HIGH RADIAL-X
C                   BOUNDARY OF A COOLANT.
C        POINTS11   A GAP HAS BEEN SPECIFIED ON THE LOW RADIAL-X
C                   BOUNDARY OF A COOLANT.
C        POINTS12   A GAP HAS BEEN SPECIFIED ON THE HIGH AXIAL-Z
C                   BOUNDARY OF A COOLANT.
C        POINTS13   A GAP HAS BEEN SPECIFIED ON THE LOW AXIAL-Z
C                   BOUNDARY OF A COOLANT.
C        POINTS14   A GAP HAS BEEN SPECIFIED ON THE HIGH THETA-Y
C                   BOUNDARY OF A COOLANT.
C        POINTS15   A GAP HAS BEEN SPECIFIED ON THE LOW THETA-Y
C                   BOUNDARY OF A COOLANT.
C     ==================================================================
C     ==================================================================
C     ERASE THE BLOCK COMPLETENESS TABLE
      DO 5 I=1,IMAX
      DO 5 J=1,JMAX
      DO 5 K=1,KMAX
      MT(I,J,K)=0
    5 CONTINUE
```

```
C     SET THE MESH RIBS EQUAL TO 1
      DO 500 L=1,JMAX
      MT(1,L,1)=1
      MT(IMAX,L,1)=1
      MT(1,L,KMAX)=1
  500 MT(IMAX,L,KMAX)=1
      DO 525 L=1,IMAX
      MT(L,1,1)=1
      MT(L,JMAX,1)=1
      MT(L,1,KMAX)=1
  525 MT(L,JMAX,KMAX)=1
      DO 550 L=1,KMAX
      MT(1,1,L)=1
      MT(IMAX,1,L)=1
      MT(1,JMAX,L)=1
  550 MT(IMAX,JMAX,L)=1
C======================================================================
C     ASSIGN THE RADIAL-X BLOCKS AND FIND THE RADIAL-X GAPS
      CALL BOUNDA(RL,RP,IM,IGR,RBL,RBH,RDG,IL,IH,NRG,1)
C     ARE THERE TOO MANY RADIAL-X GAPS
      IF(NRG.GT.MAXRG) CALL ERROR(7HPOINTS2)
C======================================================================
C     ASSIGN THE AXIAL-Z BLOCKS AND FIND THE AXIAL-Z GAPS
      CALL BOUNDA(ZL,ZP,JM,JGZ,ZBL,ZBH,ZDG,JL,JH,NZG,2)
C     ARE THERE TOO MANY AXIAL-Z GAPS
      IF(NZG.GT.MAXZG) CALL ERROR(7HPOINTS3)
C======================================================================
C     ASSIGN THE THETA-Y BLOCKS AND FIND THE THETA-Y GAPS
      CALL BOUNDA(TL,TP,KM,KGT,TBL,TBH,TDG,KL,KH,NTG,3)
C     ARE THERE TOO MANY THETA-Y GAPS
      IF(NTG.GT.MAXTG) CALL ERROR(7HPOINTS4)
C======================================================================
C     INITIALIZE ARRAYS.
      DO 6 IG=1,MAXRG
      DO 6 J=1,MAXZP
      DO 6 K=1,MAXTP
      RBBIL(IG,J,K)=0.0
      RBBIH(IG,J,K)=0.0
      GAPR(IG,J,K)=0.0
      MATRG(IG,J,K)=0
    6 CONTINUE
      DO 7 JG=1,MAXZG
      DO 7 I=1,MAXRP
      DO 7 K=1,MAXTP
      ZBBIL(JG,I,K)=0.0
      ZBBTH(JG,I,K)=0.0
```

```
      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
      GAPT(KG,I,J)=0.0
      MATTG(KG,I,J)=0
    8 CONTINUE
C=========================================================================
C     LOOK AT EVERY BLOCK
      DO 120 L=1,LMAX
C     ASSIGN THE CORNERS OF THE CURRENT BLOCK
      ILS=IL(L)
      IHS=IH(L)
      JLS=JL(L)
      JHS=JH(L)
      KLS=KL(L)
      KHS=KH(L)
C     IS THIS BLOCK A COOLANT OR A SOLID MATERIAL? -'COOL.,SOLID'-
      IF (MB(L).LE.0) GO TO 115
C=========================================================================
C     IS A RADIAL-X GAP PRESENT? -'NO,YES'-
      IF(RDG(L).LE.0.)GO TO 30
C=========================================================================
C     ASSIGN THE RADIAL-X GAP
      IG=IGR(IHS)
      DO 20 J=JLS,JHS
      DO 10 K=KLS,KHS
C     HAS A COOLANT BEEN DEFINED ADJACENT TO THE GAP?
      IF (GAPR(IG,J,K) .GE. 0.0) GO TO 15
      CALL ERROR(8HPOINTS11)
      GO TO 100
   15 GAPR(IG,J,K)=RDG(L)
      MATRG(IG,J,K)=MGR(L)
      RBBIL(IG,J,K)=460.0
      RBBTH(IG,J,K)=460.0
   10 CONTINUE
   20 CONTINUE
C=========================================================================
C     IS AN AXIAL-Z GAP PRESENT? -'NO,YES'-
   30 IF(ZDG(L).LE.0.0)GO TO 60
C=========================================================================
C     ASSIGN THE AXIAL-Z GAP
```

```
      JG=JGZ(JHS)
      DO 50 I=ILS,IHS
      DO 40 K=KLS,KHS
C     HAS A COOLANT BEEN DEFINED ADJACENT TO THE GAP?
      IF (GAPZ(JG,I,K) .GE. 0.0) GO TO 45
      CALL ERROR(8HPOINTS13)
      GO TO 100
   45 GAPZ(JG,I,K)=ZDG(L)
      MATZG(JG,I,K)=MGZ(L)
      ZBBTL(JG,I,K)=460.0
      ZBBTH(JG,I,K)=460.0
   40 CONTINUE
   50 CONTINUE
C==========================================================================
C     IS A THETA-Y GAP PRESENT? -'NO,YES'-
   60 IF(TDG(L).LE.0.0)GO TO 100
C==========================================================================
C     ASSIGN THE THETA-Y GAP
      KG= KGT(KHS)
      DO 80 I=ILS,IHS
      DO 70 J=JLS,JHS
C     HAS A COOLANT BEEN DEFINED ADJACENT TO THE GAP?
      IF (GAPT(KG,I,J) .GE. 0.0) GO TO 75
      CALL ERROR (8HPOINTS15)
      GO TO 100
   75 GAPT(KG,I,J)=TDG(L)
      MATTG(KG,I,J)=MGT(L)
      TBBTL(KG,I,J)=460.0
      TBBTH(KG,I,J)=460.0
   70 CONTINUE
   80 CONTINUE
C==========================================================================
C/    RECORD ALL POINTS BELONGING TO THE CURRENT BLOCK -'IMPOS,'OK'-
  100 DO 110 I=ILS,IHS
      DO 110 J=JLS,JHS
      DO 110 K=KLS,KHS
      IF(MT(I,J,K).NE.0) CALL ERROR (7HPOINTS5)
      MT(I,J,K)=1
  110 CONTINUE
      GO TO 120
CE    *********************************************************************
C==========================================================================
C     ASSIGN THE COOLANT NUMBER
  115 J= IABS(MB(L))
C     ASSIGN THE FLOW DIRECTION
      IP=IABS(IPATH(J))
```

```
C      IS IT A VALID DIRECTION? -'YES,NO'-
       IF (IP.GT.0.AND.IP.LT.4) GO TO 200
C      IMPOSSIBLE FLOW DIRECTION NUMBER
       CALL ERROR(7HPOINTS6)
C      IS AN EXTERNAL COOLANT FLOWING INTO A BOUNDARY ?
C      DOES AN INTERNAL COOLANT FLOW THROUGH ONLY ONE LEVEL OF POINTS ?
   200 IF(IHS.EQ.ILS.AND.IP.EQ.1) CALL ERROR(7HPOINTS7)
       IF(JHS.EQ.JLS.AND.IP.EQ.2) CALL ERROR(7HPOINTS8)
       IF(KHS.EQ.KLS.AND.IP.EQ.3) CALL ERROR(7HPOINTS9)
C=============================================================================
C      IS THE FLOW IN THE RADIAL-X DIRECTION? -'YES,NO'-
       IF (IP.EQ.1) GO TO 220
C      SET TWO SWITCHES TO INDICATE ON WHICH SIDE OF ITS RADIAL-X
C     .BOUNDARIES THE COOLANT LIES
       IG1= IGR(IHS)
       IG2= IGR(ILS-1)
       IF (ILS.LE.1) IG2=0
       DO 210J= JLS,JHS
       DO 210K= KLS,KHS
       IF (IG2.LE.0)GO TO 180
C      IS THERE A GAP ADJACENT TO COOLANT'S LOW BOUNDARY?
       IF (GAPR(IG2,J,K) .LE. 0.0) GO TO 175
       CALL ERROR(8HPOINTS11)
       GO TO 100
   175 GAPR(IG2,J,K)=-1.0E-10+GAPR(IG2,J,K)
       RBBIL(IG2,J,K)=460.0
       RBBIH(IG2,J,K)=460.0
   180 IF (IG1.LE.0) GO TO 210
C      IS THERE A GAP ADJACENT TO COOLANT HIGH BOUNDARY?
       IF (GAPR(IG1,J,K) .LE. 0.0) GO TO 184
       CALL ERROR(8HPOINTS10)
       GO TO 100
   184 GAPR(IG1,J,K)=-2.0E-10+GAPR(IG1,J,K)
       RBBIL(IG1,J,K)=460.0
       RBBIH(IG1,J,K)=460.0
   210 CONTINUE
C=============================================================================
C      IS THE FLOW IN THE AXIAL-Z DIRECTION? -'YES,NO'-
   220 IF (IP.EQ.2) GO TO 240
C      SET TWO SWITCHES TO INDICATE ON WHICH SIDE OF ITS AXIAL-Z
C     .BOUNDARIES THE COOLANT LIES
       JG1= JGZ(JHS)
       JG2= JGZ(JLS-1)
       IF (JLS.LE.1) JG2=0
       DO 230 I=ILS,IHS
       DO 230 K= KLS,KHS
```

```
      IF (JG1.LE.0) GO TO 215
C     IS THERE A GAP ADJACENT TO COOLANT HIGH BOUNDARY?
      IF (GAPZ(JG1,I,K) .LE. 0.0) GO TO 213
      CALL ERROR(8HPOINTS12)
      GO TO 100
  213 GAPZ(JG1,I,K)=-2.0E-10+GAPZ(JG1,I,K)
      ZBBTL(JG1,I,K)=460.0
      ZBBTH(JG1,I,K)=460.0
  215 IF (JG2.LE.0) GO TO 230
C     IS THERE A GAP ADJACENT TO COOLANT'S LOW BOUNDARY?
      IF (GAPZ(JG2,I,K) .LE. 0.0) GO TO 218
      CALL ERROR(8HPOINTS13)
      GO TO 100
  218 GAPZ(JG2,I,K)=-1.0E-10+GAPZ(JG2,I,K)
      ZBBTL(JG2,I,K)=460.0
      ZBBTH(JG2,I,K)=460.0
  230 CONTINUE
C=======================================================================
C     IS THE FLOW IN THE THETA-Y DIRECTION? -'YES,NO'-
  240 IF (IP.EQ.3) GO TO 100
C     SET TWO SWITCHES TO INDICATE ON WHICH SIDE OF ITS THETA-Y
C    .BOUNDARIES THE COOLANT LIES
      KG1= KGT(KHS)
      KG2= KGT(KLS-1)
      IF (KLS.LE.1) KG2=0
      DO 250 I= ILS,IHS
      DO 250 J= JLS,JHS
      IF (KG1.LE.0) GO TO 255
C     IS THERE A GAP ADJACENT TO COOLANT HIGH BOUNDARY?
      IF (GAPT(KG1,I,J) .LE. 0.0) GO TO 253
      CALL ERROR (8HPOINTS14)
      GO TO 100
  253 GAPT(KG1,I,J)=-2.0E-10+GAPT(KG1,I,J)
      TBBTL(KG1,I,J)=460.0
      TBBTH(KG1,I,J)=460.0
  255 IF (KG2.LE.0) GO TO 250
C     IS THERE A GAP ADJACENT TO COOLANT'S LOW BOUNDARY?
      IF (GAPT(KG2,I,J) .LE. 0.0) GO TO 257
      CALL ERROR (8HPOINTS15)
      GO TO 100
  257 GAPT(KG2,I,J)=-1.0E-10+GAPT(KG2,I,J)
      TBBTL(KG2,I,J)=460.0
      TBBTH(KG2,I,J)=460.0
  250 CONTINUE
      GO TO 100
C=======================================================================
```

```
C       HAVE ALL BLOCKS BEEN CHECKED? -*YES,NOTYET*-
  120 CONTINUE
C=========================================================================
C       INITIALIZE THE MATRG,MATZG AND MATTG VALUES WHICH ARE ON GAPLINES
C       BOUNDING COOLANTS.
        DO 260 I=1,NRG
        DO 260 J=1,JMAX
        DO 260 K=1,KMAX
  260 IF(GAPR(I,J,K).LT.-.5E-10) MATRG(I,J,K)=100
        DO 270 I=1,IMAX
        DO 270 J=1,NZG
        DO 270 K=1,KMAX
  270 IF(GAPZ(I,J,K).LT.-.5E-10) MATZG(I,J,K)=100
        DO 280 I=1,IMAX
        DO 280 J=1,JMAX
        DO 280 K=1,NTG
  280 IF(GAPT(I,J,K).LT.-.5E-10) MATTG(I,J,K)=100
C=========================================================================
C/      HAVE ALL POINTS BEEN ASSIGNED TO A BLOCK? -*NO,YES*-
        DO 150 I=1,IMAX
        DO 140 J=1,JMAX
        DO 130 K=1,KMAX
        IF(MT(I,J,K).LE.0) CALL ERROR(7HPOINTS1)
  130 CONTINUE
  140 CONTINUE
  150 CONTINUE
CE      ******************************************************************
        RETURN
C=========================================================================
        END
```

```
      SUBROUTINE BOUNDA(AL,AP,M,IGA,ABL,ABH,ADG,ILA,IHA,NAG,IP)
      INCLUDE       COMDIM
C======================================================================
C     ASSIGN THE BLOCK AND GAP LOCATIONS FOR ONE DIMENSION
C     =================================================================
C     ERROR STOPS=
C        BOUNDA1    THE LOW BOUNDARY FOR A BLOCK IS TOO LARGE OR
C                   LARGER THAN THE LARGEST BOUNDARY DEFINED IN THAT
C                   DIMENSION.
C        BOUNDA2    THE HIGH BOUNDARY OF A BLOCK IS LARGER THAN THE
C                   LARGEST BOUNDARY DEFINED IN THAT DIMENSION.
C        BOUNDA3    AN INTERNAL COOLANT BLOCK HAS BEEN SPECIFIED IN
C        VIOLATION OF CODE LIMITATIONS.  MOST LIKELY ERRORS ARE
C                   1.  A NORMAL COOLANT HAS BEEN SPECIFIED FOR A
C                       PROBLEM IN CYLINDRICAL COORDINATES.
C                       2.  AN INTERNAL COOLANT HAS TWO BOUNDARIES WHICH
C                       ARE COINCIDENT.
C     =================================================================
C======================================================================
      DIMENSION        AL    (  1),AP    (  1),IGA   (  1),ABL   (  1),
     1                 ABH   (  1),ADG   (  1),ILA   (  1),IHA   (  1)
C======================================================================
C     CALCULATE THE POINT LOCATIONS
      AP(1)=AL(1)
      DO 10 I=2,M
      AP(I)=(AL(I)+AL(I-1))*0.5
   10 CONTINUE
      AP(M+1)=AL(M)
C======================================================================
C     ERASE THE GAP LOCATION ARRAY
      DO 20 I=1,M
      IGA(I)=0
   20 CONTINUE
      M1=M+1
C======================================================================
C     FIND THE BLOCK BOUNDARY INDICES IN THE CURRENT DIMENSION
C    .FOR EACH BLOCK
      DO 70 L=1,LMAX
C     IS IT A COOLANT OR A SOLID MATERIAL BLOCK? -,SOLID,COOL.,-
      IF(MB(L).GT.0)GO TO 25
C     IS IT AN EXTERNAL OR INTERNAL COOLANT? -,INTERN,EXTERN,-
      IF(ABL(L).NE.ABH(L))GO TO 25
CB    ASSIGN THE EXTERNAL COOLANT BLOCK
C     IS IT ON THE LOW INDEX OUTSIDE -,YES,NO,-
      IF(ABL(L).EQ.AP(1))GO TO 105
```

```
C      IS IT ON THE HIGH INDEX OUTSIDE -*YES,NO*-
       IF(ABL(L).EQ.AP(M1))GO TO 110
C      ILLEGAL INTERNAL COOLANT CHANNEL
       CALL ERROR (7HBOUNDA3)
C      ASSIGN THE LOW INDEX OUTSIDE COOLANT LIMITS
   105 ILA(L)=1
       IHA(L)=1
       IGA(1)=-1
       GO TO 70
C      ASSIGN THE HIGH INDEX OUTSIDE COOLANT LIMITS
   110 ILA(L)=M1
       IHA(L)=M1
       IGA(M)=-1
       GO TO 70
CE     ****************************************************************
CB     ASSIGN THE SOLID MATERIAL OR INTERNAL COOLANT BLOCK
C/     FIND THE LOW BLOCK BOUNDARY INDEX -*OK,NONE*-
    25 DO 30 I=2,M
       IF(ABL(L).LT.AP(I))GO TO 40
    30 CONTINUE
CE     ****************************************************************
C      THE LOW BLOCK BOUNDARY LIES OUTSIDE THE SYSTEM
       CALL ERROR(7HBOUNDA1)
C=======================================================================
C      ASSIGN THE LOW BLOCK BOUNDARY
    40 ILA(L)=I
C/     FIND THE HIGH BLOCK BOUNDARY INDEX -*OK,NONE*-
       I=I+1
       DO 50 J=I,M1
       IF(ABH(L).LE.AP(J))GO TO 60
    50 CONTINUE
CE     ****************************************************************
C      THE HIGH BLOCK BOUNDARY LIES OUTSIDE THE SYSTEM
       CALL ERROR(7HBOUNDA2)
C      ASSIGN THE HIGH BLOCK BOUNDARY INDEX
    60 IHA(L)=J-1
C=======================================================================
C      SET THE GAP SWITCH WHEN NECESSARY
       IF(ADG(L).GT.0.0)IGA(J-1)=-1
C      IS IT A COOLANT BLOCK -*NO,YES*-
       IF(MB(L).GT.0)GO TO 70
       N=-MB(L)
C      DOES THE COOLANT FLOW IN THE CURRENT DIRECTION  -*YES,NO*-
       IF (IP.EQ.IABS(IPATH(N))) GO TO 70
C      ASSIGN A GAP BOUNDARY ON BOTH SIDES OF THE COOLANT
       IGA(J-1)=-1
```

```
      IGA(I-2)=-1
C     HAVE ALL BLOCKS BEEN CHECKED? -,YES,NO,-
   70 CONTINUE
C=======================================================================
C/    INSPECT THE GAP INDICES AND ASSING THE NUMBER OF BOUNDARIES
      J=0
      DO 80 I=1,M
      IF(IGA(I).EQ.0)GO TO 80
      J=J+1
      IGA(I)=J
   80 CONTINUE
      NAG=J
CE    ******************************************************************
CE    ******************************************************************
      RETURN
C=======================================================================
      END
```

```
      SUBROUTINE CONSTA
      INCLUDE      COMDIM
C===========================================================================
C/    CALCULATE RADIAL-X GEOMETRIC CORRECTION FACTOR 1 (RLN TABLE C1)
C     ======================================================================
C     ======================================================================
      DO 10 I=1,IM
      DELR(I)=RP(I+1)-RP(I)
      IF(ISHAPE.EQ.1)GO TO 5
      RLN(I)=ALOG(RL(I)/RP(I))/ALOG(RP(I+1)/RP(I))
      GO TO 10
    5 RLN(I)=(RL(I)-RP(I))/DELR(I)
   10 CONTINUE
C     CALCULATE AXIAL-Z GEOMETRIC CORRECTION FACTOR 1 (ZLN TABLE C1)
      DO 20 J=1,JM
      DELZ(J)=ZP(J+1)-ZP(J)
      ZLN(J)=(ZL(J)-ZP(J))/DELZ(J)
   20 CONTINUE
C     CALCULATE THETA-Y GEOMETRIC CORRECTION FACTOR 1 (TLN TABLE C1)
      DO 30 K=1,KM
      DELT(K)=TP(K+1)-TP(K)
      TLN(K)=(TL(K)-TP(K))/DELT(K)
   30 CONTINUE
CE    **********************************************************************
C/    CALCULATE RADIAL-X GEOMETRIC GAP CORRECTION FACTORS 2 AND 3
C    .(RATIOK,RATIOH, TABLES C2,C4)
      DO 40 I=1,IM1
      IF(IGR(I).LE.0)GO TO 40
      IG=IGR(I)
      IF(ISHAPE.EQ.1)GO TO 35
      RATIOK(IG)=ALOG(RP(I+1)/RP(I))/ALOG(RP(I+1)/RL(I))
      RATIOH(IG)=RL(I)*ALOG(RP(I+1)/RL(I))
      RATIOC(IG)=1.0/(RL(I)*ALOG(RP(I+1)/RP(I)))
      GO TO 40
   35 RATIOK(IG)=DELR(I)/(RP(I+1)-RL(I))
      RATIOH(IG)=RP(I+1)-RL(I)
      RATIOC(IG)=1.0/DELR(I)
   40 CONTINUE
      RATIOK(NRG)=0.0
      RATIOH(NRG)=0.0
      IF (ISHAPE. EQ. 1) GO TO 43
      RATIOC(NRG) = 1.0/(RL(IM)*ALOG(RP(IM+1)/RP(IM)))
      GO TO 45
   43 RATIOC(NRG)=1.0/DELR(IM)
   45 CONTINUE
```

```
CE    ***********************************************************************
C/    CALCULATE AXIAL-Z GEOMETRIC GAP CORRECTION FACTORS 2 AND 3
C    .(ZATIOK,ZATIOH, TABLES C2,C4)
      DO 50 J=1,JM1
      IF(JGZ(J).LE.0)GO TO 50
      JG=JGZ(J)
      ZATIOK(JG)=DELZ(J)/(ZP(J+1)-ZL(J))
      ZATIOH(JG)=ZP(J+1)-ZL(J)
   50 CONTINUE
      ZATIOK(NZG)=0.0
      ZATIOH(NZG)=0.0
CE    ***********************************************************************
C/    CALCULATE THETA-Y GEOMETRIC GAP CORRECTION FACTORS 2 AND 3
C    .(TATIOK,TATIOH, TABLES C2,C4)
      DO 60 K=1,KM1
      IF(KGT(K).LE.0)GO TO 60
      KG=KGT(K)
      TATIOK(KG)=DELT(K)/(TP(K+1)-TL(K))
      TATIOH(KG)=TP(K+1)-TL(K)
   60 CONTINUE
      TATIOK(NTG)=0.0
      TATIOH(NTG)=0.0
CE    ***********************************************************************
CB    PASS THROUGH ALL BLOCKS TO CALCULATE THE GAP CORRECTION FACTORS
      DO 120 L=1,LMAX
C     DEFINE THE CORNERS OF THE CURRENT BLOCK
      ILS= IL(L)
      IHS= IH(L)
      JLS= JL(L)
      JHS= JH(L)
      KLS= KL(L)
      KHS= KH(L)
C     IS IT A COOLANT OR A SOLID MATERIAL BLOCK? -,COOL.,SOLID,-
      IF(MB(L).LE.0)GO TO 110
C======================================================================
C/    CALCULATE RADIAL-X GEOMETRIC GAP CORRECTION FACTOR 4
C    .(RATIOB, TABLE C3)
      IF(RDG(L).LE.0.0)GO TO 90
      IG=IGR(IHS)
      IF(ISHAPE.EQ.1)GO TO 70
      X=ALOG((RL(IHS)-RDG(L))/RP(IHS))/ALOG(RP(IHS+1)/RL(IHS))
      GO TO 75
   70 X=(RL(IHS)-RP(IHS)-RDG(L))/(RP(IHS+1)-RL(IHS))
   75 DO 80 J=JLS,JHS
      DO 80 K=KLS,KHS
      RATIOB(IG,J,K)=X
```

```
CE    ******************************************************************
C/    CALCULATE AXIAL-Z GEOMETRIC GAP CORRECTION FACTORS 2 AND 3
C    .(ZATIOK,ZATIOH, TABLES C2,C4)
      DO 50 J=1,JM1
      IF(JGZ(J).LE.0)GO TO 50
      JG=JGZ(J)
      ZATIOK(JG)=DELZ(J)/(ZP(J+1)-ZL(J))
      ZATIOH(JG)=ZP(J+1)-ZL(J)
   50 CONTINUE
      ZATIOK(NZG)=0.0
      ZATIOH(NZG)=0.0
CE    ******************************************************************
C/    CALCULATE THETA-Y GEOMETRIC GAP CORRECTION FACTORS 2 AND 3
C    .(TATIOK,TATIOH, TABLES C2,C4)
      DO 60 K=1,KM1
      IF(KGT(K).LE.0)GO TO 60
      KG=KGT(K)
      TATIOK(KG)=DELT(K)/(TP(K+1)-TL(K))
      TATIOH(KG)=TP(K+1)-TL(K)
   60 CONTINUE
      TATIOK(NTG)=0.0
      TATIOH(NTG)=0.0
CE    ******************************************************************
CB    PASS THROUGH ALL BLOCKS TO CALCULATE THE GAP CORRECTION FACTORS
      DO 120 L=1,LMAX
C     DEFINE THE CORNERS OF THE CURRENT BLOCK
      ILS= IL(L)
      IHS= IH(L)
      JLS= JL(L)
      JHS= JH(L)
      KLS= KL(L)
      KHS= KH(L)
C     IS IT A COOLANT OR A SOLID MATERIAL BLOCK? -,COOL.,SOLID,-
      IF(MB(L).LE.0)GO TO 110
C=======================================================================
C/    CALCULATE RADIAL-X GEOMETRIC GAP CORRECTION FACTOR 4
C    .(RATIOB, TABLE C3)
      IF(RDG(L).LE.0.0)GO TO 90
      IG=IGR(IHS)
      IF(ISHAPE.EQ.1)GO TO 70
      X=ALOG((RL(IHS)-RDG(L))/RP(IHS))/ALOG(RP(IHS+1)/RL(IHS))
      GO TO 75
   70 X=(RL(IHS)-RP(IHS)-RDG(L))/(RP(IHS+1)-RL(IHS))
   75 DO 80 J=JLS,JHS
      DO 80 K=KLS,KHS
      RATIOB(IG,J,K)=X
```

```
   80 CONTINUE
C=========================================================================
C      CALCULATE AXIAL-Z GEOMETRIC GAP CORRECTION FACTOR 4
C     .(ZAT10B, TABLE C3)
   90 IF(ZDG(L).LE.0.0)GO TO 100
      JG=JGZ(JHS)
      X=(ZL(JHS)-ZP(JHS)-ZDG(L))/(ZP(JHS+1)-ZL(JHS))
      DO 95 I=ILS,IHS
      DO 95 K=KLS,KHS
      ZAT10B(JG,I,K)=X
   95 CONTINUE
C=========================================================================
C      CALCULATE THETA-Y GEOMETRIC GAP CORRECTION FACTOR 4
C     .(TAT10B, TABLE C3)
  100 IF(TDG(L).LE.0.0)GO TO 120
      KG=KGT(KHS)
      X=(TL(KHS)-TP(KHS)-TDG(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=IGR(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
```

```
      SUBROUTINE GEOMET
      INCLUDE       COMDIM
C=======================================================================
CB    CALCULATE THE GEOMETRICAL FACTOR AND CELL VOLUMES FOR NODAL POINTS
C=    ==================================================================
C     ERROR STOPS
C        GEOMLT1   SOME POINT HAS A NEGATIVE CALCULATED VOLUME
C=======================================================================
C=======================================================================
      LOGICAL          ,IGAP        ,JGAP        ,KGAP
C=======================================================================
C     PASS THROUGH ALL PLANES IN THE RADIAL-X DIRECTION
      DO 400 I= 1,IM
C     IS A RADIAL-X GAP ADJACENT? -'NO,YES'-
      IF(IGR(I).EQ.0) GO TO 10
C=======================================================================
C     ASSIGN THE RADIAL-X GAP INDEX
      IG= IGR(I)
      IGAP= .TRUE.
      GO TO 20
C=======================================================================
C     SET THE RADIAL-X GAP AVERAGES TO ZERO
   10 IGAP= .FALSE.
      DGJR= 0.0
      DGKJ= 0.0
      DGR= 0.0
C=======================================================================
C     FOR A GIVEN RADIAL-X PLANE, PASS THROUGH THE AXIAL-Z COLUMNS
   20 DO 300 J= 1,JM
C     IS AN AXIAL-Z GAP ADJACENT? -'NO,YES'-
      IF (JGZ(J).EQ.0) GO TO 30
C=======================================================================
C     ASSIGN THE AXIAL-Z GAP INDEX
      JG= JGZ(J)
      JGAP= .TRUE.
      GO TO 40
C=======================================================================
C     SET THE AXIAL-Z GAP AVERAGES TO ZERO
   30 JGAP= .FALSE.
      DGIK= 0.0
      DGKI= 0.0
      DGZ= 0.0
C=======================================================================
C     FOR THE GIVEN AXIAL-Z COLUMNS, PASS THROUGH ALL THETA-Y POINTS
   40 DO 200 K= 1,KM
```

```
C     IS A THETA-Y GAP ADJACENT? -'NO,YES'-
      IF (KGT(K).EQ.0) GO TO 50
C======================================================================
C     ASSIGN THE THETA-Y GAP INDEX
      KG= KGT(K)
      KGAP= .TRUE.
      GO TO 70
C======================================================================
C     SET THE THETA-Y GAP AVERAGES TO ZERO
   50 KGAP= .FALSE.
      DGIJ= 0.0
      DGJI= 0.0
      DGT= 0.0
C======================================================================
C     IS A RADIAL-X GAP ADJACENT? -'NO,YES'-
   70 IF(.NOT.IGAP) GO TO 80
C======================================================================
C     ASSIGN THE RADIAL-X GAP AVERAGES
      DGJK= (GAPR(IG,J,K)+GAPR(IG,J+1,K))*0.5
      DGKJ= (GAPR(IG,J,K)+GAPR(IG,J,K+1))*0.5
      DGR= GAPR(IG,J,K)
C======================================================================
C     IS AN AXIAL-Z GAP ADJACENT? -'NO,YES'-
   80 IF(.NOT.JGAP) GO TO 90
C======================================================================
C     ASSIGN THE AXIAL-Z GAP AVERAGES
      DGIK=(GAPZ(JG,I,K)+GAPZ(JG,I+1,K))*0.5
      DGKI=(GAPZ(JG,I,K)+GAPZ(JG,I,K+1))*0.5
      DGZ=GAPZ(JG,I,K)
C======================================================================
C     IS A THETA-Y GAP ADJACENT? -'NO,YES'-
   90 IF(.NOT.KGAP) GO TO 100
C======================================================================
C     ASSIGN THE THETA-Y GAP ZVERAGES
      DGIJ=(GAPT(KG,I,J)+GAPT(KG,I+1,J))*0.5
      DGJI=(GAPT(KG,I,J)+GAPT(KG,I,J+1))*0.5
      DGT=GAPT(KG,I,J)
C======================================================================
C     IS THE GEOMETRY CYLINDRICAL OR RECTANGULAR? -'RECT.,CYLIN.'-
  100 RLX=RL(I-1)
      IF (I.EQ.1) RLX=0.0
      ZLX=ZL(J-1)
      IF (J .EQ. 1) ZLX=0.0
      TLX=TL(K-1)
      IF (K.EQ.1) TLX=0.0
      IF(ISHAPE.NE.0)GO TO 150
```

```
C=========================================================================
C     CALCULATE THE GEOMETRY FACTOR AND NODE VOLUME FOR A
C    .CYLINDRICAL GEOMETRY
      RR(I,J,K)=(ZL(J)-ZLX-DGIK)*(TL(K)-TLX -DGIJ )/ALOG(RP(I+1)/RP(I))
      RZ(I,J,K)=((RL(I)-DGJK)**2-RLX**2)*(TL(K)-TLX-DGJI)/(ZP(J+1)-ZP(J)
     X)*0.5
      RT(I,J,K)=(RL(I)-RLX-DGKJ)*(ZL(J)-ZLX-DGKI)/(RP(I)*(TP(K+1)-TP(K))
     X)
      V(I,J,K)=((RL(I)-DGR)**2-RLX**2)*(ZL(J)-ZLX-DGZ)*(TL(K)-TLX-DGT)
     X*0.5
      GO TO 199
C     CALCULATE THE GEOMETRY FACTOR AND NODE VOLUME FOR A
C    .RECTANGULAR GEOMETRY
  150 RR(I,J,K)=(TL(K)-TLX-DGIJ)*(ZL(J)-ZLX-DGIK)/DELR(I)
      RZ(I,J,K)=(TL(K)-TLX-DGJI)*(RL(I)-RLX-DGJK)/DELZ(J)
      RT(I,J,K)=(RL(I)-RLX-DGKJ)*(ZL(J)-ZLX-DGKI)/DELT(K)
      V(I,J,K)=(TL(K)-TLX-DGT)*(RL(I)-RLX-DGR)*(ZL(J)-ZLX-DGZ)
  199 IF (V(I,J,K) .LT. 0.0) CALL ERROR (7HGEOMET1)
  200 CONTINUE
CE    ********************************************************************
  300 CONTINUE
  400 CONTINUE
C=========================================================================
C     CLEAN-UP CALCULATED ARRAYS SO THAT ONLY PHYSICALLY MEANINGFUL
C     DATA IS PRESENTED
      DO 1000 I=1,IMAX
      DO 1000 K=1,KMAX
      RZ(I,JMAX,K)=0.0
      RR(I,1,K)=0.0
      RR(I,JMAX,K)=0.0
      RT(I,1,K)=0.0
      RT(I,JMAX,K)=0.0
      V(I,1,K)=0.0
      V(I,JMAX,K)=0.0
 1000 CONTINUE
      DO 1500 J=1,JMAX
      DO 1500 K=1,KMAX
      RZ(IMAX,J,K)=0.0
      RZ(1,J,K)=0.0
      RT (IMAX,J,K)=0.0
      RT(1,J,K)=0.0
      RR(IMAX,J,K)=0.0
      V(1,J,K)=0.0
      V(IMAX,J,K)=0.0
 1500 CONTINUE
      DO 2000 I=1,IMAX
```

```
      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 DISTAN
     -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 DISTAN
     -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 DISTAN
     -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)
      INCLUDE COMDIM
C=========================================================================
CB    PRINT A THREE-DIMENSIONAL ARRAY
C=========================================================================
C=========================================================================
C=========================================================================
      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/
C=========================================================================
C     WRITE THE PROPER HEADING
      NADD=10
      IF (IFY.EQ.1) NADD=25
      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
      N=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
      N=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
      N=IMAX
   40 DO 600 M=1,MM
      NUM2=0
  100 NUM1=NUM2+1
      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)
```

```
  221 WRITE (6,170)
  170 FORMAT (1H0)
      GO TO (200,300,400),NPRINT
C=======================================================================
C     OUTPUT PRINT IN RADIAL-X DIRECTION, THETA-Y DIRECTION HORIZONTAL,
C     AXIAL-Z DIRECTION VERTICAL
C=======================================================================
  200 DO 420 J=1,JMAX
      GO TO (421,422,423),IFY
  421 WRITE (6,203) J,(IX(M,J,K),K=NUM1,NUM2)
  203 FORMAT (I4,1X,I6,24I5)
      GO TO 420
  422 WRITE (6,201) J,(X(M,J,K),K=NUM1,NUM2)
  201 FORMAT (I4,1P10E12.4)
      GO TO 420
  423 WRITE (6,202) J,(X(M,J,K),K=NUM1,NUM2)
  202 FORMAT (I4,1P10E12.5)
  420 CONTINUE
      IF(NUM2 .LT. N) GO TO 100
      GO TO 600
C=======================================================================
C     OUTPUT PRINT IN AXIAL-Z DIRECTION, RADIAL-X DIRECTION HORIZONTAL,
C     THETA-Y DIRECTION VERTICAL
C=======================================================================
  300 DO 320 K=1,KMAX
      GO TO (321,322,323),IFY
  321 WRITE (6,203) K,(IX(I,M,K), I=NUM1,NUM2)
      GO TO 320
  322 WRITE (6,201) K,(X(I,M,K), I=NUM1,NUM2)
      GO TO 320
  323 WRITE (6,202) K,(X(I,M,K), I=NUM1,NUM2)
  320 CONTINUE
      IF(NUM2 .LT. N) GO TO 100
      GO TO 600
C=======================================================================
C     OUTPUT PRINT IN THETA-Y DIRECTION, RADIAL-X DIRECTION HORIZONTAL,
C     AXIAL-Z DIRECTION VERTICAL
C=======================================================================
  400 DO 230 J=1,JMAX
      GO TO (231,232,233), IFY
  231 WRITE (6,203) J,(IX(I,J,M),I=NUM1,NUM2)
      GO TO 230
  232 WRITE (6,201) J,( X(I,J,M),I=NUM1,NUM2)
      GO TO 230
  233 WRITE (6,202) J,( X(I,J,M),I=NUM1,NUM2)
  230 CONTINUE
      IF(NUM2 .LT. N) GO TO 100
  600 CONTINUE
      RETURN
C=======================================================================
      END
```

```
      SUBROUTINE DPRINT
      INCLUDE      COMDIM
C======================================================================
CB    PRINT THE POINT AND BOUNDARY ASSIGNMENTS
C======================================================================
C======================================================================
C/    PRINT THE RADIAL-X POINT, GRIDLINE, AND BOUNDARY ASSIGNMENTS
      WRITE (6,100) (ZA(I), I=1,12), PNAME(1), PNAME(6), PNAME(7),
     -PNAME(6), PNAME(7)
  100 FORMAT(1H1,30X,12A6,///,24X,A6,23H  BOUNDARY  ASSIGNMENTS,//,3X,
     -8HSEQUENCE,9X,5HPOINT,9X,9HGRID LINE,6X,14HCOOLANT OR GAP/4X,
     -6HNUMBER,8X,8HLOCATION,8X,8HLOCATION,7X,15HBOUNDARY NUMBER/
     -18X,2A6,4X,2A6/)
      DO 20 I=1,IMAX
      TEM=RP(I)*12.0
      WRITE (6,107)I,TEM
      IF (I.EQ.IMAX) GO TO 20
      TEM=RL(I)*12.0
      IF (IGR(I).GT.0) GO TO 10
      WRITE (6,106) I,TEM
      GO TO 20
   10 IG=IGR(I)
      WRITE(6,105)I,TEM,IG
   20 CONTINUE
CE    *****************************************************************
C     PRINT THE AXIAL-Z POINT, GRID LINES AND BOUNDARY ASSIGNMENTS
      WRITE (6,100) (ZA(I), I=1,12), PNAME(2), PNAME(6), PNAME(7),
     -PNAME(6), PNAME(7)
      DO 40J= 1,JMAX
      TEM= ZP(J)*12.0
      WRITE (6,107) J,TEM
      IF (J.EQ.JMAX) GO TO 40
      TEM= ZL(J)*12.0
      IF (JGZ(J).GT.0) GO TO 30
      WRITE (6,106) J,TEM
      GO TO 40
   30 JG= JGZ(J)
      WRITE (6,105)J,TEM,JG
   40 CONTINUE
CE    *****************************************************************
C/    PRINT THE THETA-Y POINT AND BOUNDARY ASSIGNMENTS
      WRITE (6,100) (ZA(I),I=1,12),PNAME(3),PNAME(4),PNAME(5),PNAME(4),
     -PNAME(5)
      DO 60 K=1,KMAX
      TEM= TP(K)*SCALE
```

```
      WRITE (6,107) K,TEM
      IF (K.EQ.KMAX) GO TO 60
      TEM= TL(K)*SCALE
      IF (KGT(K).GT.0) GO TO 50
      WRITE (6,106) K,TEM
      GO TO 60
   50 KG= KGT(K)
      WRITE (6,105)K,TEM,KG
   60 CONTINUE
CE    ****************************************************************
      RETURN
C=====================================================================
  105 FORMAT (3X,I6,16X,F16.4,I16)
  106 FORMAT (3X,I6,16X,F16.4)
  107 FORMAT (3X,I3, 3X,F16.4)
C=====================================================================
      END
```

```
      SUBROUTINE INITEM
      INCLUDE        COMDIM
C=======================================================================
CB    ASSIGN THE INITIAL TEMPERATURE DISTRIBUTION
C=======================================================================
C         INITEM1   THE TEMPERATURE BLOCK'S LOWER RADIAL-X BOUNDARY DOES
C                   NOT COINCIDE WITH ANY OF THE RADIAL-X GRID BOUNDARIES
C         INITEM2   THE TEMPERATURE BLOCK'S UPPER RADIAL-X BOUNDARY DOES
C                   NOT COINCIDE WITH ANY OF THE RADIAL-X GRID BOUNDARIES
C         INITEM3   THE TEMPERATURE BLOCK'S LOWER AXIAL-Z BOUNDARY DOES
C                   NOT COINCIDE WITH ANY OF THE AXIAL-Z GRID BOUNDARIES.
C         INITEM4   THE TEMPERATURE BLOCK'S UPPER AXIAL-Z BOUNDARY DOES
C                   NOT COINCIDE WITH ANY OF THE AXIAL-Z GRID BOUNDARIES.
C         INITEM5   THE TEMPERATURE BLOCK'S LOWER THETA-Y BOUNDARY DOES
C                   NOT COINCDE WITH ANY OF THE THETA-Y GRID BOUNDARIES.
C         INITEM6   THE TEMPERATURE BLOCK'S UPPER THETA-Y BOUNDARY DOES
C                   NOT COINCIDE WITH ANY OF THE THETA-Y GRID BOUNDARIES.
C         INITEM7   AN INITIAL TEMPERATURE HAS NOT BEEN ASSIGNED TO SOME
C                   INTERNAL POINT.
C     =================================================================
C=======================================================================
C     SET ALL INTERNAL TEMPERATURES TO ZERO ABSOLUTE
      DO 10 I=2,IM
      DO 10 J=2,JM
      DO 10 K=2,KM
      TT(I,J,K)=0.0
   10 CONTINUE
C=======================================================================
C     SET ALL EXTERNAL RADIAL-X COOLANT TEMPERATURES TO ZERO FAHRENHEIT
      DO 20 J=1,JMAX
      DO 20 K=1,KMAX
      TT(1,J,K)=460.0
      TT(IMAX,J,K)=460.0
   20 CONTINUE
C=======================================================================
C     SET ALL EXTERNAL AXIAL-Z COOLANT TEMPERATURES TO ZERO FAHRENHEIT
      DO 30 I=1,IMAX
      DO 30 K=1,KMAX
      TT(I,1,K)=460.0
      TT(I,JMAX,K)=460.0
   30 CONTINUE
C=======================================================================
C     SET ALL EXTERNAL THETA-Y COOLANT TEMPERATURES TO ZERO FAHRENHEIT
      DO 40 I=1,IMAX
      DO 40 J=1,JMAX
```

```
      TT(I,J,1)=460.0
      TT(I,J,KMAX)=460.0
   40 CONTINUE
C=========================================================================
C=========================================================================
C     READ AND CONVERT THE GEOMETRIC DATA OF A TEMPERATURE BLOCK
   45 READ(5,100)RMIN,RMAX,ZMIN,ZMAX,TMIN,TMAX,FLAG1,FLAG2
C     HAVE ALL INITIAL TEMPERATURES BEEN READ? -'YES,NO'-
      IF (RMIN+RMAX+ZMIN+ZMAX+TMIN+TMAX.EQ.0.0) GO TO 1000
      CALL FCARD (6.0,RMIN,RMAX,ZMIN,ZMAX,TMIN,TMAX,FLAG1,FLAG2)
C=========================================================================
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=========================================================================
CB    ASSIGN THE UPPER AND LOWER RADIAL-X LIMITS OF THE TEMPERATURE BLOC
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    *******************************************************************
C     THE LOW RADIAL-X BOUNDARY LIES OUTSIDE THE SYSTEM
      CALL ERROR(7HINITEM1)
C=========================================================================
C     ASSIGN THE INDEX OF THE LOW RADIAL-X TEMPERATURE BLOCK BOUNDARY
   60 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    *******************************************************************
C     THE HIGH RADIAL-X BOUNDARY LIES OUTSIDE THE SYSTEM
      CALL ERROR(7HINITEM2)
C=========================================================================
C     ASSIGN THE INDEX OF THE HIGH RADIAL-X TEMPERATURE BLOCK BOUNDARY
   80 IMA=I
C     ADJUST THE APPROPRIATE INDEX IF THE BLOCK DESCRIBES AN EXTERNAL
C    .COOLANT
      IF (IMA.EQ.1) IMI=1
```

```
      IF (IMI.EQ.IMAX) IMA=IMAX
CE    ******************************************************************
CB    ASSIGN THE UPPER AND LOWER AXIAL-Z LIMITS OF THE TEMPERATURE BLOCK
C/    FIND THE INDEX OF THE LOW AXIAL-Z TEMPERATURE BLOCK BOUNDARY
      DO 110 J=1,JM
      IF(ZMIN.LE.1.00001*ZL(J).AND.ZMIN.GE.0.99999*ZL(J))GO TO 120
  110 CONTINUE
CE    ******************************************************************
C     THE LOW AXIAL-Z BOUNDARY LIES OUTSIDE THE SYSTEM
      CALL ERROR(7HINITEM3)
C=======================================================================
C     ASSIGN THE INDEX OF THE LOW AXIAL-Z TEMPERATURE BLOCK BOUNDARY
  120 JMI=J+1
      J1=J
C/    FIND THE INDEX OF THE HIGH AXIAL-Z TEMPERATURE BLOCK BOUNDARY
      DO 130 J=J1,JM
      IF(ZMAX.LE.1.00001*ZL(J).AND.ZMAX.GE.0.99999*ZL(J))GO TO 140
  130 CONTINUE
CE    ******************************************************************
C     THE HIGH AXIAL-Z BOUNDARY LIES OUTSIDE THE SYSTEM
      CALL ERROR(7HINITEM4)
C     ASSIGN THE INDEX OF THE HIGH AXIAL-Z TEMPERATURE BLOCK BOUNDARY
  140 JMA=J
C     ADJUST THE APPROPRIATE INDEX IF THE BLOCK DESCRIBES AN EXTERNAL
C    .COOLANT
      IF (JMA.EQ.1) JMI=1
      IF (JMI.EQ.JMAX) JMA=JMAX
CE    ******************************************************************
CB    ASSIGN THE UPPER AND LOWER THETA-Y LIMITS OF THE TEMPERATURE BLOCK
C/    FIND THE INDEX OF THE LOW THETA-Y TEMPERATURE BLOCK BOUNDARY
      DO 150 K=1,KM
      IF(TMIN.LE.1.00001*TL(K).AND.TMIN.GE.0.99999*TL(K))GO TO 160
  150 CONTINUE
CE    ******************************************************************
C     THE LOW THETA-Y BOUNDARY LIES OUTSIDE THE SYSTEM
      CALL ERROR(7HINITEM5)
C=======================================================================
C     ASSIGN THE INDEX OF THE LOW THETA-Y TEMPERATURE BLOCK BOUNDARY
  160 KMI=K+1
      K1=K
C/    FIND THE INDEX OF THE HIGH THETA-Y TEMPERATURE BLOCK BOUNDARY
      DO 170 K=K1,KM
      IF(TMAX.LE.1.00001*TL(K).AND.TMAX.GE.0.99999*TL(K))GO TO 180
  170 CONTINUE
CE    ******************************************************************
C     THE HIGH THETA-Y BOUNDARY LIES OUTSIDE THE SYSTEM
```

```
      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+460.0
C=======================================================================
C     ASSIGN THE TEMPERATURES TO ALL POINTS IN THE TEMPERATURE BLOCK
      DO 200 I=IMI,IMA
      DO 200 J=JMI,JMA
      DO 200 K=KMI,KMA
      IT(I,J,K)=TEM
  200 CONTINUE
      GO TO 45
C=======================================================================
C=======================================================================
C/    HAS A TEMPERATURE BEEN ASSIGNED TO EACH INTERNAL POINT? -'NO,YES.-
 1000 NC=NC+1
      CARD(NC)=0.0
      DO 1010 I=2,IM
      DO 1010 J=2,JM
      DO 1010 K=2,KM
      IF(IT(I,J,K).EQ.0.0) GO TO 1100
 1010 CONTINUE
CE    ******************************************************************
C     PUT THE INTIAL 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(E12.4,60X,A6,A2)
C=======================================================================
      END
```

```
      SUBROUTINE FLOWCA
      INCLUDE        COMDIM
C     ===============================================================
C     READ AND PRINT THE COOLANT DATA CARDS
C=====================================================================
C     ERROR STOPS=
C        FLOWCA1    THE REYNOLDS NUMBER LIMITS ARE NOT IN SEQUENCE
C        FLOWCA2    THE FLOWRATE LIMITS ARE NOT IN SEQUENCE
C        FLOWCA3    THE INLET TEMPERATURE LIMITS ARE NOT IN SEQUENCE
C=====================================================================
C=====================================================================
      DIMENSION       CDIRCT(  2)
      DATA CDIRCT     /6HPOSITI, 6HNEGATI/
C=====================================================================
      WRITE (6,120)(ZA(I), I = 1,12)
  120 FORMAT (1H1,30X,12A6,///,50X,23HCOOLANT  SPECIFICATIONS)
C     MAKE ALL COOLANTS NON-EXISTING
      DO 5 I=1,MAXFLO
      FLOW(I)= 0.0
    5 CONTINUE
C     READ THE COOLANT INFORMATION
   10 READ (5,100) DA,D1,D2,D3,D4,FLAG1,FLAG2
      N=DA+0.1
C     HAVE ALL COOLANT DATA CARDS BEEN READ? -'YES,NO'-
      IF(N.LE.0) GO TO 291
      CALL FCARD (5,0,DA,D1,D2,D3,D4,FLAG1,FLAG2,DUM)
C     RECORD THE EXISTENCE OF THE CURRENT COOLANT
      FLOW(N)= 1.0
      WRITE (6,101) N
  101 FORMAT (///,27H SPECIFICATIONS FOR COOLANT,I3,/)
      IP = IABS(IPATH(N))
      DIRECT = CDIRCT(1)
      IF (IPATH(N) .GT. 0) GO TO 300
      DIRECT = CDIRCT(2)
  300 WRITE (6,302) DIRECT, PNAME(IP)
  302 FORMAT(31H THE COOLANT IS FLOWING IN THE ,A6,3HVE ,A6,10H DIRECTIO
     -N)
C     ***************************************************************
C     READ THE REMAINING SPECIFICATIONS OF THIS COOLANT
      RLIM1(N)=D1
      RLIM2(N)=D2
      RLIM3(N)=D3
      RLIM4(N)=D4
C/    CHECK TO SEE THAT THE REYNOLDS NUMBER LIMITS ARE IN SEQUENCE
      IF(RLIM2(N).LT.RLIM1(N)) CALL ERROR (7HFLOWCA1)
```

```
      IF(RLIM3(N).LT.RLIM2(N)) CALL ERROR (7HFLOWCA1)
      IF(RLIM4(N).LT.RLIM3(N)) CALL ERROR (7HFLOWCA1)
CE    ******************************************************************
      WRITE(6,105)D1,D2,D3,D4
      READ  (5,100)FLODEP(N),FLIM1(N),FLIM2(N),FLIM3(N),       FLIM4(N)
      CALL FCARD (5.0,FLODEP(N),FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N),
     XFLAG1,FLAG2,DUM)
C/    CHECK TO SEE THAT THE FLOWRATE LIMITS ARE IN SEQUENCE
      IF(FLIM2(N).LT.FLIM1(N))CALL ERROR (7HFLOWCA2)
      IF(FLIM3(N).LT.FLIM2(N))CALL ERROR (7HFLOWCA2)
      IF(FLIM4(N).LT.FLIM3(N))CALL ERROR (7HFLOWCA2)
CE    ******************************************************************
      READ  (5,100)TINDEP(N),TLIM1(N),TLIM2(N),TLIM3(N),       TLIM4(N)
      CALL FCARD (5.0,TINDEP(N),TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N),
     XFLAG1,FLAG2,DUM)
C/    CHECK TO SEE THAT THE INLET TEMPERATURE LIMITS ARE IN SEQUENCE
      IF(TLIM2(N).LT.TLIM1(N))CALL ERROR(7HFLOWCA3)
      IF(TLIM3(N).LT.TLIM2(N))CALL ERROR(7HFLOWCA3)
      IF(TLIM4(N).LT.TLIM3(N))CALL ERROR(7HFLOWCA3)
CE    ******************************************************************
      IFLO(N)=FLODEP(N)+1.01
      ITIN(N)=TINDEP(N)+1.01
      I=IFLO(N)
CB    PRINT THE FLOWRATE AND INLET TEMPERATURE DEPENDENCES
C     DETERMINE THE DEPENDENCE OF THE FLOWRATE -'NONE,TIME,OUTLET,INLET'
      GO TO (20,21,22,23),I
C     PRINT THAT THERE ARE NO STEPCHANGES IN FLOWRATE
   20 WRITE(6,106)
      GO TO 30
C     PRINT THAT TIME DEFINES THE STEPCHANGES IN FLOWRATE
   21 WRITE(6,107)FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N)
      GO TO 30
C     PRINT THAT THE OUTLET TEMPERATURE DEFINES THE STEPCHANGES IN
C    *FLOWRATE
   22 WRITE(6,108)FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N)
      GO TO 30
C     PRINT THAT THE INLET TEMPERATURE DEFINES THE STEPCHANGES IN
C    *FLOWRATE
   23 WRITE(6,109)FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N)
C/    DETERMINE THE DEPENDENCE OF THE INLET TEMPERATURE
C    *-'NONE,TIME,FLOW,OUTLET'-
   30 I=ITIN(N)
      GO TO (31,32,33,34),I
CE    ******************************************************************
C     PRINT THAT THERE ARE NO STEPCHANGES IN THE INLET TEMPERATURE
   31 WRITE(6,110)
```

```
      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      FLOW1000
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
      CARD(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 (68H OUTLET TEMPERATURE(F) DEFINES THE STEP CHANGES IN INLEFLOW1480
   XT TEMPERATURE,1P4E14.4)                                            FLOW1490
C=====================================================================FLOW1500
      END                                                              FLOW1510
```

```
      SUBROUTINE TIME
      INCLUDE      COMDIM
C=======================================================================
C     READ AND PRINT THE TIME STEPS
C=======================================================================
C     ERROR STOPS=
C        TIME1      THE FIRST TIME CARD DOES NOT SPECIFY TIME UNITS.
C                   THE PROBLEM IS NOT AFFECTED IF THE STEADY STATE
C                   OPTION IS BEING USED.
C        TIME2      TOO MANY TIME HISTORY CARDS HAVE BEEN READ.
C        TIME3      THE INPUT CONTAINS NO TIME HISTORY CARDS AND THE
C                   STEADY STATE OPTION IS NOT BEING USED.
C     =================================================================
C=======================================================================
      DIMENSION        PTIME ( 20,4)
      EQUIVALENCE      (RCP    ,PTIME )
C=======================================================================
C=======================================================================
      M=0
C     READ THE NEXT TIMESTEP
    2 M=M+1
      READ(5,100)FTIME(M),DTIME(M),ST,FT,FLAG1,FLAG2
  100 FORMAT (4E12.4,24X,A6,A2)
      IF (DTIME(M) .LE. 0.0) GO TO 185
      CALL FCARD (4.0,FTIME(M),DTIME(M),ST,FT,FLAG1,FLAG2,DUM,DUM)
      ITAPE(M)=ST
      IUNIT=FT
C/    ARE THE UNITS THE SAME AS ON THE LAST CARD? -'NO,YES,WRONG'-
    4 IF(IUNIT)5,5,7
    5 IF(M-1)6,6,61
CE    ******************************************************************
C     THE FIRST TIMECARD HAS NO UNITS
    6 CALL ERROR (5HTIME1)
C/    CONVERT THE TIMES TO THE PROPER UNITS
    7 IF(IUNIT-2)8,9,60
    8 D=3600.0
      GO TO 61
    9 D=60.0
CE    ******************************************************************
C     CONVERT FTIME AND DTIME TO HOURS.
   61 DTIME(M)=DTIME(M)/D
      FTIME(M)=FTIME(M)/D
C     COMPENSATE FOR THE ROUNDOFF ERROR IN DTIME.
   60 DTIME(M)=DTIME(M)+TIMERR(DTIME(M))
C     HAVE ALL TIMECARDS BEEN READ? -'NO,YES'-
```

```
      GO TO 2
  185 IF(.NOT.SW(10).AND.M.EQ.1) CALL ERROR(5HTIME3)
      NC=NC+1
      CARD(NC)=0.0
      MMAX=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,18HEND OF TIM
     -E 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,I14)
      IF(.NOT.SW(10)) RETURN
      WRITE(6,240)
  240 FORMAT(1H0, 88H   THE ABOVE TIME HISTORY WILL BE IGNORED BECAUSE T
     .HE STEADY STATE OPTION IS BEING USED.)
  250 DO 260 M=1,21
      DTIME(M)=0.0
      FTIME(M)=0.0
  260 ITAPE(M)=0
      MMAX=1
      RETURN
C     *****************************************************************
C     *****************************************************************
C     *****************************************************************
      FUNCTION TIMERR(TIM)
C     =================================================================
C     DETERMINE MAXIMUM ROUNDOFF ERROR INCURRED IN CONVRTING TIME SCALF
C     DATA TO HOURS.
C     =================================================================
      REAL              MAG
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
      TIMERR=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
      TIMERR=    MAG(16+KX)                                             TIME1060
C     ==================================================================TIME1070
      RETURN                                                            TIME1080
      END                                                               TIME1090
```

```
      SUBROUTINE EXTRA
      INCLUDE      COMDIM
C=======================================================================
C     READ THE EXTRA FUNCTION COEFFICIENTS AND PRINT THEM
C=======================================================================
C=======================================================================
      DIMENSION       A      (18)
      EQUIVALENCE     (A(1),A1)
C=======================================================================
      READ (5,100) A1,A2,A3,A4,A5,A6,FLAG1,FLAG2
      CALL FCARD (6.0,A1,A2,A3,A4,A5,A6,FLAG1,FLAG2)
      READ (5,100) A7,A8,A9,A10,A11,A12,FLAG1,FLAG2
      CALL FCARD (6.0,A7,A8,A9,A10,A11,A12,FLAG1,FLAG2)
      READ (5,100) A13,A14,A15,A16,A17,A18,FLAG1,FLAG2
      CALL FCARD (6.0,A13,A14,A15,A16,A17,A18,FLAG1,FLAG2)
      DO 110 I=1,18
      IF (A(I) .EQ. 0.0) A(I)=0.0
  110 CONTINUE
      WRITE(6,101) (ZA(I), I = 1,12)
      WRITE (6,102) (I,A(I), I=1,18)
C     SPACE OVER BLANK CARD
      READ (5,100) X
      NC=NC+1
      CARD(NC)=0.0
C=======================================================================
  100 FORMAT (6E12.4,A6,A2)
  101 FORMAT(1H1,30X,12A6,///,50X,28HFUNCTION  CONTROL  CONSTANTS,//,6X,
     -15HCONSTANT NUMBER,12X,5HVALUE/)
  102 FORMAT (1H0,I14,12X,1PE14.6)
C=======================================================================
      RETURN
      END
```

```
      SUBROUTINE PRETEM
      INCLUDE       COMDIM
C=====================================================================
C=====================================================================
C*    ERROR STOP:
C*    PRETEM1       THE PROBLEM SIZE DOES NOT MATCH THE INITIAL
C*                  TEMPERATURE DISTRIBUTION DATA
C=====================================================================
      EQUIVALENCE       (TEMP   ,W      )
      INTEGER           TT
      DIMENSION         NUMBER( 50),TEMP   (  1)
      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/
C=====================================================================
      WRITE (6,1105) (ZA(I), I = 1,12)
 1105 FORMAT (1H1,30X,12A6,///,46X,32HINPUT  TEMPERATURE  DISTRIBUTION)
C=====================================================================
      CURTI=0.0
      NITER=0
C=====================================================================
C     READ THE FIRST CARD WITH THE GRID SIZE
      READ (5,260) CURTI, BIMAX,BJMAX,BKMAX,BNITER,FLAG1,FLAG2
      IAMAX=BIMAX +0.1
      JAMAX=BJMAX +0.1
      KAMAX=BKMAX +0.1
      NITER = BNITER +0.1
C     IS A PREVIOUSLY CALCULATED TEMPERATURE DISTRIBUTION PROVIDED?
C    .-'NO,YES'-
      IF (IAMAX .NE. 0) GO TO 105
      NC=NC+1
      CARD(NC)=0.0
      GO TO 38
C=====================================================================
C/    IS THE GRID SIZE CORRECT?  -'NO,YES'-
  105 IF (IAMAX.NE.IMAX) GO TO 11
      IF (JAMAX.NE.JMAX) GO TO 11
      IF (KAMAX.EQ.KMAX) GO TO 21
CE    *****************************************************************
C     THE TEMPERATURE INPUT DOES NOT MATCH THE PROBLEM
   11 CALL ERROR (7HPRETEM1)
C=====================================================================
C     READ THE INITIAL TEMPERATURE DISTRIBUTION
   21 CALL FCARD (5,0,CURTI,BIMAX,BJMAX,BKMAX,BNITER,FLAG1,FLAG2,DUM)
      NT=IMAX*JMAX*KMAX
```

```
      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 NUM
     1BER,4X,5HINLET,5X,6HOUTLET,10X,14HCOOLANT NUMBER,4X,5HINLET,5X,6HO
     2UTLET)
      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,I16,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
      CURTI=0.0
      NITER=0
  185 X1=CURTI*60.0
      X2=CURTI*3600.0
      WRITE (6,250) CURTI,X1,X2,NITER
  250 FORMAT (1H0,21H  THE CURRENT TIME IS,F10.4,11H HOURS  =  ,F10.4,13
     1H MINUTES  =  ,F13.5,8H SECONDS,4X,I4,31H ITERATIONS HAVE BEEN PER
     2FORMED,//)
C=====================================================================
C     CONVERT TEMPERATURES TO INTEGRAL OUTPUT UNITS
   38 DO 15 I=1,IMAX
```

```
      DO 15 K=1,KMAX
      DO 15 J=1,JMAX
      TT(I,J,K) = T(I,J,K) - 459.5
   15 CONTINUE
C     SET MESH RIBS 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 (E12.0,4F12.1,12X,A6,A2)
  122 FORMAT(////)
C*    ******************************************************************
C*    ******************************************************************
      SUBROUTINE RCARD (TR,LENGTH)
C=    ==================================================================
      DIMENSION        TR     (  1),TC     (  6)
C=    ==================================================================
      N=LENGTH/6
      NLEFT=LENGTH-N*6
      N2=0
      DO 100 I=1,N
      N1=N2+1
      N2=N2+6
      READ (5,110) (TR(J),J=N1,N2),FLAG1,FLAG2
      CALL FCARD (6.0,TR(N1),TR(N1+1),TR(N1+2),TR(N1+3),TR(N1+4),TR(N1+5
     1),FLAG1,FLAG2)
  100 CONTINUE
      IF (NLEFT.EQ.0) GO TO 130
```

```
      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+I                                                              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         /6HMP2  1, 6H      /
C     ==================================================================
      CS1=0.0
      CS2=0.0
      DP=.FALSE.
      IWNS=-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,MMAX
      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=MMAX
```

```
  150 FTIME(K+1)=FTIME(K)
      DTIME(K+1)=DTIME(K)
      ITAPE(K+1)=ITAPE(K)
      K=K-1
      IF (K.GE.I) GO TO 150
      FTIME(I)=CURTI+1.0E-10/3600.0
      DTIME(I)=1.0E-10/3600.0
      ITAPE(I)=0
      MMAX=MMAX+1
      GO TO 165
C=    ==================================================================
  160 FIRST=.FALSE.
      IF(.NOT.SW(10)) GO TO 165
      IF(CS1.LT.1.0) GO TO 162
      GO TO 165
  162 CALL STEADY
      IWNS=NITER+19
C     START THE NEXT TIME PERIOD
  165 M=M+1
      CTI=CURTI
      IC=1
C     HAVE ALL THE TIME PERIODS BEEN FINISHED
      IF (M.GT.MMAX) GO TO 200
      ENDTI=FTIME(M)
      DTSTAN=DTIME(M)
      NTSTAN=ITAPE(M)
      DT=DTSTAN
      NTA=NTSTAN
      IF (NTSTAN.EQ.0) NTA=10000
C=    ==================================================================
CB    CALCULATE THE TEMPERATURES DURING ONE SERIES OF TIMESTEPS
C    .-'END,NEXT'-
C     HAS COMPUTER TIME ALMOST RUN OUT? -'NO,YES'-
  170 IF (WARN(I)) 181, 200, 200
C     HAS NUMBER OF PRINTED PAGES SPECIFIED ALMOST RUN OUT? -'NO,YES'-
  181 CALL PAGLIM(4,201,5)
C     DEFINE THE TIME IN  THE MIDDLE OF THE TIMESTEP
  180 DATI=CURTI+DTSTAN/2.0
C     SAVE TEMPERATURE RESULTS OF NEXT TO LAST SMOOTHING ITERATION FOR
C     CALCULATING RESIDUALS(STEADY STATE OPTION ONLY).
      IF(.NOT.SW(10)) GO TO 182
      IF(NITER.NE.IWNS) GO TO 182
      REWIND NSTORE
      WRITE (NSTORE)(((T(I,J,K),I=1,IQ),J=1,JQ),K=1,KQ)
C     CALCULATE THE MATERIAL PROPERTIES
  182 CALL BLOCK
```

```
C     CALCULATE THE AVERAGE CONDUCTIVITIES BETWEEN POINTS               MP2  920
      CALL CONDUC                                                       MP2  930
C     CALCULATE THE NEXT TEMPERATURE DISTRIBUTION                       MP2  940
      NOTEST=.TRUE.                                                     MP2  950
      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=CTI+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)
      INCLUDE       COMDIM
C======================================================================
C======================================================================
C======================================================================
C======================================================================
      DIMENSION        IB     (  2)
C======================================================================
      IERROR(1) = IB(1)
      IERROR(2) = IB(2)
C     RECORD THE ERROR
      WRITE(6,100) IERROR
  100 FORMAT(10H ERROR AT ,2A6)
C     IS THE DUMP SWITCH ON? -'YES,NO'.
C     GIVE A DUMP
      IF (SW(1)) CALL DUMP
      STOP
C======================================================================
      END
```

```
      SUBROUTINE STEADY
      INCLUDE       COMDIM
C     ===================================================================
C     CALCULATE THE STEADY STATE TEMPERATURES.
C     ===================================================================
      LOGICAL            ENDSW, LREAD, WRES, NOTEST, FIR2
      DIMENSION          XT(IQ,JQ,KQ), STEADY(IQ,JQ,KQ)
      EQUIVALENCE        (XT,TT), (RCP,STEADY)
C     ===================================================================
C     ===================================================================
C     SET CONSTANTS AND INDICATORS.
C     ===================================================================
      DATI = -1.0E+8
      TOL=1.0E-4
      RIFAC=1.50
      DTMAX=3.0
      ITMAX=1000
      IPSET=10
      MSTORE=2
      NSTORE=3
      LSTORE=4
C     SET AN INDICATOR TO SUPRESS ANY FURTHER CALCULATION OF REAL
C     CAPACITANCE TERMS.
      CS2=2.0
C     ===================================================================
C     INITIALIZATIONS
C     ===================================================================
      CURTI=0.0
      NITER=0
      IDT=1
      NSETT=0
      N=0
      NT=0
      DTFAC=1.0
      FIR2=.FALSE.
      NADV=1
      LREAD=.TRUE.
      ENDSW=.FALSE.
      WRES=.FALSE.
C     ===================================================================
C     PREPARE TO DO A SET OF ITERATIONS.
C     ===================================================================
C     DETERMINE THE PARAMETERS WHICH CONTROL THE TIME STEP MAGNITUDE
C     AND RATE OF INCREASE DURING THE CURRENT ITERATION SET.
  230 IF(IDT.EQ.2) GO TO 250
```

```
      NADV=1
      IF(FIR2) NADV=0
      GO TO 270
  250 NADV=0
      DTFAC=DTFAC/RIFAC
      FIRST=.TRUE.
C     IF INSTABILITIES OCCURRED DURING THE PRECEEDING ITERATION,AND IF
C     THIS IS NOT THE FIRST ITERATION SET,RECOVER FROM DRUM THE
C     TEMPERATURE RESULTS OF THE NEXT TO LAST SUCCESSFUL SET.
      ISTORE=MSTORE
      IF(LREAD) ISTORE=LSTORE
      CALL STORE(ISTORE,1)
      GO TO 270
  260 DTFAC=DTFAC/RIFAC
      NADV=0
C     IF INSTABILITY OCCURS DURING THE FIRST ITERATION SET,RECOVER THE
C     INITIAL TEMPERATURES FROM DRUM.
      CALL STORE(MSTORE,1)
C     ==================================================================
C     PERFORM A SET OF ITERATIONS.
C     ==================================================================
  270 DO 1000 M=1,IPSET
C     DETERMINE THE TIME STEP FOR EACH ITERATION.DO NOT ALLOW THE TIME
C     STEP TO EXCEED DTMAX HOURS.
      XN=N
      DTSTAN=DTFAC*(.10*XN+1.0)
      IF(DTSTAN.GT.DTMAX ) DTSTAN=DTMAX
      DT=DTSTAN
C     DETERMINE THE CONDUCTANCES AND PSEUDO CAPACITANCES FOR THE
C     CURRENT ITERATION.
      CALL BLOCK
      CALL CONDUC
      CALL STACAP
C     IF THIS IS THE FIRST ITERATION OF A PROBLEM,SAVE THE INITIAL
C     TEMPERATURES ON DRUM.
      IF(NT.NE.0) GO TO 300
      CALL STORE(MSTORE,2)
C     ADVANCE THE ITERATION COUNTERS.
  300 N=N+NADV
      NT=NT+1
C     CALCULATE TEMPERATURES FOR THE CURRENT ITERATION.
      IF(WARN(I).LT.0.0.AND.NT.LT.ITMAX) GO TO 400
      ENDSW=.TRUE.
      NOTEST=.TRUE.
      CALL STEP($410,NOTEST)
      GO TO 410
```

```
  400 NOTEST=.FALSE.                                                     STEA 920
      CALL STEP($220,NOTEST)                                             STEA 930
  410 CALL COOL                                                          STEA 940
      CALL SORT                                                          STEA 950
      IF(ENDSW) 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                                       STEA1000
      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
      NSETT=NSETT+1                                                      STEA1050
C     ===================================================================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
C     ===================================================================STEA1140
      IF(LREAD) GO TO 305                                                STEA1150
      ISTORE=MSTORE                                                      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
C     ===================================================================STEA1230
C     ENTER FROM SUBROUTINE PRINT TO EVALUATE RESIDUALS FOR THE FINAL    STEA1240
C     SMOOTHING ITERATION.                                               STEA1250
C     ===================================================================STEA1260
      ENTRY RESID                                                        STEA1270
      ARLSE=.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
C     ===================================================================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                                                    STEA1380
      DO 1500 J = 2,JM                                                  STEA1390
      DO 1500 K=2,KM                                                    STEA1400
      IF (RCP(I,J,K) .GT. 1.0E10) GO TO 1500                            STEA1410
      CAP = (RCP(I,J,K) * (T(I,J,K) - XT(I,J,K)))/DTSTAN                STEA1420
      TCENT = ( T(I,J,K) + XT(I,J,K))/2.0                               STEA1430
      AA1= TCENT -((T(I+1,J,K) + XT(I+1,J,K))/2.0)                      STEA1440
      AA2=((T(I-1,J,K) + XT(I-1,J,K))/2.0) - TCENT                      STEA1450
      AA3= TCENT -((T(I,J+1,K) + XT(I,J+1,K))/2.0)                      STEA1460
      AA4=((T(I,J-1,K) + XT(I,J-1,K))/2.0) - TCENT                      STEA1470
      AA5=TCENT - ((T(I,J,K+1) + XT(I,J,K+1))/2.0)                      STEA1480
      AA6=((T(I,J,K-1) + XT(I,J,K-1))/2.0) - TCENT                      STEA1490
      H1=-KR(I,J,K)*KR(I,J,K)*AA1                                       STEA1500
      H2=KR(I-1,J,K)*KR(I-1,J,K)*AA2                                    STEA1510
      H3=-KZ(I,J,K)*KZ(I,J,K)*AA3                                       STEA1520
      H4=KZ(I,J-1,K)*KZ(I,J-1,K)*AA4                                    STEA1530
      H5=-KT(I,J,K)*KT(I,J,K)*AA5                                       STEA1540
      H6=KT(I,J,K-1)*KT(I,J,K-1)*AA6                                    STEA1550
      VERG      = H1 + H2 + H3 + H4 + H5 + H6 + W(I,J,K)                STEA1560
      AVERG= ABS(H1) + ABS(H2) + ABS(H3) + ABS(H4) + ABS(H5) + ABS(H6)  STEA1570
     . + ABS(W(I,J,K))                                                  STEA1580
      ACAP = ABS(CAP)                                                   STEA1590
C     CALCULATE THE CONVERGENCE PARAMETER AT EVERY MATERIAL POINT IN    STEA1600
C     THE PROBLEM.                                                      STEA1610
      STEADY(I,J,K) = ACAP/AVERG                                        STEA1620
 1500 CONTINUE                                                          STEA1630
      IF(ENDSW.OR.WRES) GO TO 360                                       STEA1640
C     ==================================================================STEA1650
C     COMPARE THE CONVERGENCE PARAMETER AT EVERY MATERIAL POINT         STEA1660
C     AGAINST A SPECIFIED MAXIMUM ALLOWABLE VALUE TO TEST WHETHER OR    STEA1670
C     NOT THE STEADY STATE SOLUTION HAS BEEN ATTAINED.                  STEA1680
C     ==================================================================STEA1690
  330 DO 200 I=2,IM                                                     STEA1700
      DO 200 J=2,JM                                                     STEA1710
      DO 200 K=2,KM                                                     STEA1720
      IF(STEADY(I,J,K).GT.1.0E10) GO TO 200                             STEA1730
      IF(STEADY(I,J,K).GE.TOL) GO TO 210                                STEA1740
  200 CONTINUE                                                          STEA1750
C     ==================================================================STEA1760
C     PREPARE TO RETURN TO SUBROUTINE MP2 AND PERFORM THE SMOOTHING     STEA1770
C     ITERATIONS.                                                       STEA1780
C     ==================================================================STEA1790
  205 CALL STACAP                                                       STEA1800
      DTIME(2)=DTSTAN/10.0                                              STEA1810
      FTIME(2)=DTIME(2)*20.9                                            STEA1820
      ITAPE(2)=7                                                        STEA1830
```

```
      NITER=NT
C     PRINT THE STEADY STATE TEMPERATURE RESULTS BEFORE RETURNING
C     TO MP2.
      CALL PRINT
C     SET AN INDICATOR TO SHOW THAT THE STEADY STATE ITERATIONS HAVE
C     BEEN COMPLETED.
      CS1=2.0
C     ==================================================================
C     RETURN TO MP2 AND PERFORM THE SMOOTHING ITERATIONS.
C     ==================================================================
      RETURN
C     ==================================================================
C     SET THE INDICATOR IDT TO SHOW THE MODE OF DETERMINING THE NEXT
C     VALUE OF THE TIME STEP AND START THE NEXT SET OF ITERATIONS.
C     ==================================================================
  210 IDT=1
      GO TO 230
C     IF INSTABILITIES OCCURRED IN SOLVING FOR TEMPERATURES,THE RETURN
C     FROM SUBROUTINE STEP WILL BE TO THE FOLLOWING STATEMENT.
  220 IDT=2
      IF(NSETT.NE.0) GO TO 230
      IDT=3
      GO TO 260
C     ==================================================================
C     PRINT TEMPERATURES AND/OR RESIDUALS AT THE END OF THE PROBLEM.
C     ==================================================================
C     PREPARE THE ARRAY OF CONVERGENCE PARAMETERS AT THE END OF THE
C     PROBLEM(I.E.RESIDUALS)FOR PRINTING.
  300 DO 350 I=1,IMAX
      DO 350 J=1,JMAX
      DO 350 K=1,KMAX
      IF(STEADY(I,J,K).GT.1.0E10) STEADY(I,J,K)=0.0
  350 CONTINUE
C     PRINT THE LAST AVAILABLE TEMPERATURE RESULTS IF THE PROBLEM IS TO
C     TERMINATE IN THIS SUBROUTINE.
      IF(NRES) GO TO 435
      NITER=NT
      CALL TEMPS
C     PRINT THE RESIDUALS AT THE END OF THE PROBLEM.
  435 WRITE(6,440)
  440 FORMAT(1H0,56X,9HRESIDUALS)
      CALL ARRAY(IDUM,STEADY,2)
      IF(NRES) RETURN
C     PUNCH THE LAST AVAILABLE TEMPERATURE RESULTS IF THE PROBLEM IS TO
C     TERMINATE IN THIS SUBROUTINE.
      CALL PUN
```

```
      CALL MERR
C     ==================================================================
C     ******************************************************************
C     ******************************************************************
C     ******************************************************************
C     ==================================================================
      SUBROUTINE STACAP
C     ==================================================================
C     SET CAPACITANCE TERMS FOR MAXIMUM STABILITY.
C     ==================================================================
      DO 80 I=2,IM
      DO 80 J=2,JM
      DO 80 K=2,KM
      IF(RCP(I,J,K).GT.1.0E10) GO TO 80
      RCP(I,J,K) = RR(I-1,J,K)*KR(I-1,J,K) + RR(I,J,K)*KR(I,J,K) +
     .             RZ(I,J-1,K)*KZ(I,J-1,K) + RZ(I,J,K)*KZ(I,J,K) +
     .             RT(I,J,K-1)*KT(I,J,K-1) + RT(I,J,K)*KT(I,J,K)
   80 CONTINUE
      RETURN
C     ==================================================================
C     ******************************************************************
C     ******************************************************************
C     ******************************************************************
C     ==================================================================
      SUBROUTINE STORE(ISTORE,NRW)
C     ==================================================================
C     WRITE ARRAYS ON DRUM OR READ ARRAYS FROM DRUM.
C     ==================================================================
      REWIND ISTORE
      GO TO (100,200), NRW
  100 READ  (ISTORE)               (TI(N), N = 1,NQ), (TO(N), N = 1,NQ),
     X(((T(I,J,K), I=1,IQ), J=1,JQ), K=1,KQ), (((RBBTL(I,J,K), I=1,IGQ),
     XJ=1,JQ), K=1,KQ),  (((RBBTH(I,J,K), I=1,IGQ), J=1,JQ), K=1,KQ),
     X(((ZBBTL(I,J,K), I=1,JGQ), J=1,IQ), K=1,KQ), (((ZBBTH(I,J,K),
     XI=1,JGQ), J=1,IQ), K=1,KQ), (((TBBTL(I,J,K), I=1,KGQ), J=1,IQ),
     XK=1,JQ), (((TBBTH(I,J,K), I=1,KGQ), J=1,IQ), K=1,JQ)
      GO TO 300
C     ==================================================================
  200 WRITE (ISTORE)               (TI(N), N = 1,NQ), (TO(N), N = 1,NQ),
     X(((T(I,J,K), I=1,IQ), J=1,JQ), K=1,KQ), (((RBBTL(I,J,K), I=1,IGQ),
     XJ=1,JQ), K=1,KQ),  (((RBBTH(I,J,K), I=1,IGQ), J=1,JQ), K=1,KQ),
     X(((ZBBTL(I,J,K), I=1,JGQ), J=1,IQ), K=1,KQ), (((ZBBTH(I,J,K),
     XI=1,JGQ), J=1,IQ), K=1,KQ), (((TBBTL(I,J,K), I=1,KGQ), J=1,IQ),
     XK=1,JQ), (((TBBTH(I,J,K), I=1,KGQ), J=1,IQ), K=1,JQ)
  300 RETURN
C     ==================================================================
      END
```

```
      SUBROUTINE BLOCK
      INCLUDE      COMDIM
C     ==================================================================
C     EVALUATE THE MATERIAL PROPERTIES OF EACH BLOCK
C     ==================================================================
C     ==================================================================
C     TURN THE GAS CONDUCTIVITY SWITCH OFF
      GAS= .FALSE.
C     EVALUATE THE PROPERTIES OF ALL BLOCKS
      DO 20 L=1,LMAX
      NI= MB(L)
C     CALCULATE THE LIMITS OF BLOCK L
C     ==================================================================
C     POINT LIMITS
      IHS= IH(L)
      ILS= IL(L)
      JHS= JH(L)
      JLS= JL(L)
      KHS= KH(L)
      KLS= KL(L)
C     BOUNDING GAPLINES
      IGLS= IGR(ILS-1)
      IGHS=IGR(IHS)
      JGLS= JGZ(JLS-1)
      JGHS= JGZ(JHS)
      KGLS= KGT(KLS-1)
      KGHS= KGT(KHS)
C     ==================================================================
C     IS THIS BLOCK A COOLANT  -'YES,NO'-
      IF (NI.LE.0) GO TO 10
C     CALCULATE THE PROPERTIES OF THE MATERIAL IN BLOCK L.
      CALL MADATA
      GO TO 20
C     ==================================================================
C     CORRECT THE GAPLINE ASSIGNMENT OF THE EXTERNAL COOLANTS
   10 IF(IHS.EQ.IMAX) IGHS=IGLS
      IF (ILS .EQ. 1) IGLS=IGHS
      IF (JHS .EQ. JMAX) JGHS=JGLS
      IF (JLS .EQ. 1) JGLS=JGHS
      IF (KHS .EQ. KMAX) KGHS=KGLS
      IF (KLS .EQ. 1) KGLS=KGHS
C     CALCULATE THE PROPERTIES OF THE COOLANT IN BLOCK L
      CALL FLUDAT(L)
C     ARE ALL BLOCKS EVALUATED  -'YES,NO'-
   20 CONTINUE
```

```
C     ================================================================
C     ISOLATE THE CONDUCTIVITIES OF POINTS ADJACENT COOLANT BOUNDARIES,
C    .STORE THEM IN THE REM-RBBT, ZEM-ZBBT, TEM-TBBT LOCATIONS FOR
C    .COOLANT BLOCK SURFACES.
C     ================================================================
C     LOCATE THE RADIAL-X COOLANT BOUNDARIES.
      DO 220 I=1,IM
      IG=IGR(I)
C     EXCLUDE THE NON-GAPLINES.
      IF(IG.EQ.0)GO TO 220
      DO 210 J=2,JM
      DO 210 K=2,KM
C     EXCLUDE THE POINTS WHERE A GAP IS ADJACENT TO THE GAPLINE.
      IF(GAPR(IG,J,K).GE.0.0)GO TO 210
C     EXCLUDE THE POINTS WHERE TWO COOLANTS ARE ADJACENT ON THE GAPLINE.
      IF(GAPR(IG,J,K).LT.(-2.5E-10))GO TO 210
C     IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE? -'NO,YES'-
      IF(GAPR(IG,J,K).LT.(-1.5E-10))GO TO 200
      REML(IG,J,K)=CONR(I,J,K)
      RBBTL(IG,J,K)=CONR(I+1,J,K)
      GO TO 210
  200 REMH(IG,J,K)=CONR(I+1,J,K)
      RBBTH(IG,J,K)=CONR(I,J,K)
  210 CONTINUE
  220 CONTINUE
C     ================================================================
C     LOCATE THE AXIAL-Z COOLANT BOUNDARIES.
      DO 250 J=1,JM
      JG=JGZ(J)
C     EXCLUDE THE NON-GAPLINES.
      IF(JG.EQ.0)GO TO 250
      DO 240 I=2,IM
      DO 240 K=2,KM
C     EXCLUDE THE POINTS WHERE A GAP IS ADJACENT TO THE GAPLINE.
      IF(GAPZ(JG,I,K).GE.0.0)GO TO 240
C     EXCLUDE THE POINTS WHERE TWO COOLANTS ARE ADJACENT ON THE GAPLINE.
      IF(GAPZ(JG,I,K).LT.(-2.5E-10))GO TO 240
C     IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE? -'NO,YES'-
      IF(GAPZ(JG,I,K).LT.(-1.5E-10))GO TO 230
      ZEML(JG,I,K)=CONZ(I,J,K)
      ZBBTL(JG,I,K)=CONZ(I,J+1,K)
      GO TO 240
  230 ZEMH(JG,I,K)=CONZ(I,J+1,K)
      ZBBTH(JG,I,K)=CONZ(I,J,K)
  240 CONTINUE
  250 CONTINUE
```

```
C     ==============================================================BLOC 920
C     LOCATE THE THETA-Y COOLANT BOUNDARIES.                         BLOC 930
      DO 300 K=1,KM                                                  BLOC 940
      KG=KGT(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.     BLOC1000
      IF (GAPT(KG,I,J) .GE. 0.0) GO TO 290                           BLOC1010
C     EXCLUDE THE POINTS WHERE TOO COOLANTS ARE ADJACENT ON THE GAPLINE.BLOC1020
      IF (GAPT(KG,I,J) .LT. (-2.5E-10))GO TO 290                     BLOC1030
C     IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE? -'NO,YES,-  BLOC1040
      IF (GAPT(KG,I,J) .LT. (-1.5E-10)) GO TO 280                    BLOC1050
      TEML(KG,I,J)=CONT(I,J,K)                                       BLOC1060
      TBBIL(KG,I,J)=CONT(I,J,K+1)                                    BLOC1070
      GO TO 290                                                      BLOC1080
  280 TEMH(KG,I,J)=CONT(I,J,K+1)                                     BLOC1090
      TBBIH(KG,I,J)=CONT(I,J,K)                                      BLOC1100
  290 CONTINUE                                                       BLOC1110
  300 CONTINUE                                                       BLOC1120
C     ==============================================================BLOC1130
      RETURN                                                         BLOC1140
      END                                                            BLOC1150
```

```
      SUBROUTINE MADATA
      INCLUDE      COMDIM
C======================================================================
C     SUPPLY THE MATERIAL THERMAL PHYSICAL PROPERTIES
C======================================================================
C======================================================================
C======================================================================
C=    =================================================================
C=    =================================================================
C=    =================================================================
C=    =================================================================
C=    =================================================================
C     IS THE MATERIAL SOLID OR GAS IN A GAP?  -'GAS,SOLID'-
   98 IF (GAS) GO TO 1300
C======================================================================
CB    CALCULATE THE SOLID MATERIAL PROPERTIES
C/    CALCULATE THE HEATGENERATION RATE FOR EACH POINT IN THE BLOCK
C======================================================================
  100 HR=DATI
      DO 150 K = KLS, KHS
      FTT = TP(K)
      DO 150 J=JLS,JHS
      FTZ = ZP(J)
      DO 150 I=ILS,IHS
      FTR=RP(I)
      DR = T(I,J,K)
      GO TO ( 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111,
     1         112, 113, 114, 115),NI
  101 X=HEAT 1(Y)
      GO TO 140
  102 X=HEAT 2(Y)
      GO TO 140
  103 X=HEAT 3(Y)
      GO TO 140
  104 X=HEAT 4(Y)
      GO TO 140
  105 X=HEAT 5(Y)
      GO TO 140
  106 X=HEAT 6(Y)
      GO TO 140
  107 X=HEAT 7(Y)
      GO TO 140
  108 X=HEAT 8(Y)
      GO TO 140
  109 X=HEAT 9(Y)
```

```
      GO TO 140
  110 X=HLAT10(Y)
      GO TO 140
  111 X=HLAT11(Y)
      GO TO 140
  112 X=HLAT12(Y)
      GO TO 140
  113 X=HLAT13(Y)
      GO TO 140
  114 X=HLAT14(Y)
      GO TO 140
  115 X=HLAT15(Y)
  140 W(I,J,K)=X*V(I,J,K)
  150 CONTINUE
CE    ******************************************************************
C/    CALCULATE THE RADIAL CONDUCTIVITY
C===================================================================
  200 DO 250 K = KLS, KHS
      FTT=TP(K)
      DO 250 I=ILS,IHS
      FTR=RP(I)
      DO 250 J=JLS,JHS
      FTZ=ZP(J)
      DR = T(I,J,K)
      GO TO ( 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211,
     1        212, 213, 214, 215),NI
  201 X=RCON 1(Y)
      GO TO 247
  202 X=RCON 2(Y)
      GO TO 247
  203 X=RCON 3(Y)
      GO TO 247
  204 X=RCON 4(Y)
      GO TO 247
  205 X=RCON 5(Y)
      GO TO 247
  206 X=RCON 6(Y)
      GO TO 247
  207 X=RCON 7(Y)
      GO TO 247
  208 X=RCON 8(Y)
      GO TO 247
  209 X=RCON 9(Y)
      GO TO 247
  210 X=RCON10(Y)
      GO TO 247
```

```
  211 X=RCON11(Y)
      GO TO 247
  212 X=RCON12(Y)
      GO TO 247
  213 X=RCON13(Y)
      GO TO 247
  214 X=RCON14(Y)
      GO TO 247
  215 X=RCON15(Y)
  247 IF(X.EQ.0.0)X=1.0E-6
      CONK(I,J,K)=1.0/X
  250 CONTINUE
CE    ************************************************************************
C/    CALCULATE THE AXIAL CONDUCTIVITY
C=============================================================================
  300 DO 350 K = KLS, KHS
      FTT=TP(K)
      DO 350 I=ILS,IHS
      FTR=RP(I)
      DO 350 J=JLS,JHS
      FTZ=ZP(J)
      DR = T(I,J,K)
      GO TO ( 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311,
     1        312, 313, 314, 315),NI
  301 X=ACON 1(Y)
      GO TO 347
  302 X=ACON 2(Y)
      GO TO 347
  303 X=ACON 3(Y)
      GO TO 347
  304 X=ACON 4(Y)
      GO TO 347
  305 X=ACON 5(Y)
      GO TO 347
  306 X=ACON 6(Y)
      GO TO 347
  307 X=ACON 7(Y)
      GO TO 347
  308 X=ACON 8(Y)
      GO TO 347
  309 X=ACON 9(Y)
      GO TO 347
  310 X=ACON10(Y)
      GO TO 347
  311 X=ACON11(Y)
      GO TO 347
```

```
  312 X=ACON12(Y)
      GO TO 347
  313 X=ACON13(Y)
      GO TO 347
  314 X=ACON14(Y)
      GO TO 347
  315 X=ACON15(Y)
  347 IF(X.EQ.0.0)X=1.0E-6
      CON2(I,J,K)=1.0/X
  350 CONTINUE
CE    ***************************************************************************
C/    CALCULATE THE THETA CONDUCTIVITY
C==================================================================================
 1500 DO 1550 K = KLS, KHS
      FTT=TP(K)
      DO 1550 I = ILS, IHS
      FTR=RP(I)
      DO 1550 J = JLS, JHS
      FTZ=ZP(J)
      DR = T(I,J,K)
      GO TO (1501, 1502, 1503, 1504, 1505, 1506, 1507, 1508, 1509, 1510,
     11511, 1512, 1513, 1514, 1515), NI
 1501 X=TCON 1(Y)
      GO TO 1547
 1502 X=TCON 2(Y)
      GO TO 1547
 1503 X=TCON 3(Y)
      GO TO 1547
 1504 X=TCON 4(Y)
      GO TO 1547
 1505 X=TCON 5(Y)
      GO TO 1547
 1506 X=TCON 6(Y)
      GO TO 1547
 1507 X=TCON 7(Y)
      GO TO 1547
 1508 X=TCON 8(Y)
      GO TO 1547
 1509 X=TCON 9(Y)
      GO TO 1547
 1510 X=TCON10(Y)
      GO TO 1547
 1511 X=TCON11(Y)
      GO TO 1547
 1512 X=TCON12(Y)
      GO TO 1547
```

```
 1513 X=TCON13(Y)                                                           MADA1840
      GO TO 1547                                                            MADA1850
 1514 X=TCON14(Y)                                                           MADA1860
      GO TO 1547                                                            MADA1870
 1515 X=TCON15(Y)                                                           MADA1880
 1547 IF(X.EQ.0.0)X=1.0E-6                                                  MADA1890
      CONT(I,J,K)=1.0/X                                                     MADA1900
 1550 CONTINUE                                                              MADA1910
CE    ******************************************************************MADA1920
C/    CALCULATE THE SPECIFIC HEAT                                           MADA1930
C===================================================================MADA1940
      IF(CS2.GT.1.0) GO TO 499                                              MADA1950
  400 DO 450 K = KLS, KHS                                                   MADA1960
      DO 450 I=ILS,IHS                                                      MADA1970
      DO 450 J=JLS,JHS                                                      MADA1980
      DR = T(I,J,K)                                                         MADA1990
      GO TO ( 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411,        MADA2000
     1         412, 413, 414, 415),NI                                       MADA2010
  401 X=SPEC 1(Y)                                                           MADA2020
      GO TO 440                                                             MADA2030
  402 X=SPEC 2(Y)                                                           MADA2040
      GO TO 440                                                             MADA2050
  403 X=SPEC 3(Y)                                                           MADA2060
      GO TO 440                                                             MADA2070
  404 X=SPEC 4(Y)                                                           MADA2080
      GO TO 440                                                             MADA2090
  405 X=SPEC 5(Y)                                                           MADA2100
      GO TO 440                                                             MADA2110
  406 X=SPEC 6(Y)                                                           MADA2120
      GO TO 440                                                             MADA2130
  407 X=SPEC 7(Y)                                                           MADA2140
      GO TO 440                                                             MADA2150
  408 X=SPEC 8(Y)                                                           MADA2160
      GO TO 440                                                             MADA2170
  409 X=SPEC 9(Y)                                                           MADA2180
      GO TO 440                                                             MADA2190
  410 X=SPEC10(Y)                                                           MADA2200
      GO TO 440                                                             MADA2210
  411 X=SPEC11(Y)                                                           MADA2220
      GO TO 440                                                             MADA2230
  412 X=SPEC12(Y)                                                           MADA2240
      GO TO 440                                                             MADA2250
  413 X=SPEC13(Y)                                                           MADA2260
      GO TO 440                                                             MADA2270
  414 X=SPEC14(Y)                                                           MADA2280
      GO TO 440                                                             MADA2290
```

```
  415 X=SPEC15(Y)
  440 RCP(I,J,K)=X*V(I,J,K)
      IF(.NOT.SW(10)) GO TO 450
      IF(X.GT.9.999E5) RCP(I,J,K)=1.0E12
  450 CONTINUE
CE    ****************************************************************
C/    CALCULATE THE RADIAL THERMAL EMISSIVITY ON THE LOW INDEX
C    .SIDE OF THE BLOCK
C=====================================================================
  499 IF (IGLS . LE . 0) GO TO 599
  500 DO 550 K = KLS, KHS
      DO 550 J=JLS,JHS
      IF (GAPR(IGLS,J,K) . EQ . 0.0) GO TO 550
      FTT=TP(K)
      FTZ=ZP(J)
      GR=HGBTL(IGLS,J,K)
      GO TO ( 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511,
     1        512, 513, 514, 515),NI
  501 X=EMRL 1(Y)
      GO TO 547
  502 X=EMRL 2(Y)
      GO TO 547
  503 X=EMRL 3(Y)
      GO TO 547
  504 X=EMRL 4(Y)
      GO TO 547
  505 X=EMRL 5(Y)
      GO TO 547
  506 X=EMRL 6(Y)
      GO TO 547
  507 X=EMRL 7(Y)
      GO TO 547
  508 X=EMRL 8(Y)
      GO TO 547
  509 X=EMRL 9(Y)
      GO TO 547
  510 X=EMRL10(Y)
      GO TO 547
  511 X=EMRL11(Y)
      GO TO 547
  512 X=EMRL12(Y)
      GO TO 547
  513 X=EMRL13(Y)
      GO TO 547
  514 X=EMRL14(Y)
      GO TO 547
```

```
  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
C=======================================================================MADA2830
  599 IF (IGHS . LE . 0) GO TO 699                                      MADA2840
  600 DO 650 K = KLS, KHS                                               MADA2850
      DO 650 J=JLS,JHS                                                  MADA2860
      IF (GAPR(IGHS,J,K) . EQ . 0.0) GO TO 650                          MADA2870
      FTT=TP(K)                                                         MADA2880
      FTZ=ZP(J)                                                         MADA2890
      DR=RBBTH(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
      REMH(IGHS,J,K)=1.0/X
  650 CONTINUE
CE    ****************************************************************
C/    CALCULATE THE AXIAL THERMAL EMISSIVITY ON THE LOW INDEX
C    .SIDE OF THE BLOCK
C=====================================================================
  699 IF (JGLS . LE . 0) GO TO 799
  700 DO 750 K = KLS, KHS
      DO 750 I = ILS, IHS
      IF (GAPZ(JGLS,I,K)  . EQ . 0.0) GO TO 750
      FTT=TP(K)
      FTR=RP(I)
      DR = ZBBTL(JGLS,I,K)
      GO TO ( 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711,
     1        712, 713, 714, 715),NI
  701 X=EMAL 1(Y)
      GO TO 747
  702 X=EMAL 2(Y)
      GO TO 747
  703 X=EMAL 3(Y)
      GO TO 747
  704 X=EMAL 4(Y)
      GO TO 747
  705 X=EMAL 5(Y)
      GO TO 747
  706 X=EMAL 6(Y)
      GO TO 747
  707 X=EMAL 7(Y)
      GO TO 747
  708 X=EMAL 8(Y)
      GO TO 747
  709 X=EMAL 9(Y)
      GO TO 747
  710 X=EMAL10(Y)
      GO TO 747
  711 X=EMAL11(Y)
      GO TO 747
  712 X=EMAL12(Y)
      GO TO 747
  713 X=EMAL13(Y)
      GO TO 747
  714 X=EMAL14(Y)
      GO TO 747
  715 X=EMAL15(Y)
  747 IF(X.EQ.0.0)X=1.0E-6
```

```
      ZEML(JGLS,I,K)=1.0/X
  750 CONTINUE
CE    ************************************************************************
C/    CALCULATE THE AXIAL THERMAL EMISSIVITY ON THE HIGH INDEX
C    .SIDE OF THE BLOCK
C=========================================================================
  799 IF (JGHS . LE . 0 ) GO TO 899
  800 DO 850 K = KLS, KHS
      DO 850 I = ILS, IHS
      IF (GAPZ(JGHS, I,K) . EQ . 0.0) GO TO 850
      FTT=TP(K)
      FIR=RP(I)
      DR = ZBBTH (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
```

```
  850 CONTINUE
CE    ***********************************************************************
C/    CALCULATE THE THETA THERMAL EMISSIVITY ON THE LOW INDEX
C    .SIDE OF THE BLOCK
C======================================================================
  899 IF( KGLS . LE . 0 )GO TO 999
  900 DO 950 I = ILS, IHS
      DO 950 J = JLS, JHS
      IF (GAPT(KGLS,I,J) . EQ . 0.0) GO TO 950
      FTR=RP(I)
      FTZ=ZP(J)
      JR = IBMTL(KGLS,I,J)
      GO TO (901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911,
     1912, 913, 914, 915), NI
  901 X=EMTL 1(Y)
      GO TO 947
  902 X=EMTL 2(Y)
      GO TO 947
  903 X=EMTL 3(Y)
      GO TO 947
  904 X=EMTL 4(Y)
      GO TO 947
  905 X=EMTL 5(Y)
      GO TO 947
  906 X=EMTL 6(Y)
      GO TO 947
  907 X=EMTL 7(Y)
      GO TO 947
  908 X=EMTL 8(Y)
      GO TO 947
  909 X=EMTL 9(Y)
      GO TO 947
  910 X=EMTL10(Y)
      GO TO 947
  911 X=EMTL11(Y)
      GO TO 947
  912 X=EMTL12(Y)
      GO TO 947
  913 X=EMTL13(Y)
      GO TO 947
  914 X=EMTL14(Y)
      GO TO 947
  915 X=EMTL15(Y)
  947 IF(X.EQ.0.0)X=1.0E-6
      TLML(KGLS,I,J)=1.0/X
  950 CONTINUE
```

```
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 = TBBTH(KGHS,I,J)                                                  MADA4700
      GO TO (1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, 1009,          MADA4710
     11010, 1011, 1012, 1013, 1014, 1015), NI                               MADA4720
 1001 X=EMTH 1(Y)                                                           MADA4730
      GO TO 1047                                                            MADA4740
 1002 X=EMTH 2(Y)                                                           MADA4750
      GO TO 1047                                                            MADA4760
 1003 X=EMTH 3(Y)                                                           MADA4770
      GO TO 1047                                                            MADA4780
 1004 X=EMTH 4(Y)                                                           MADA4790
      GO TO 1047                                                            MADA4800
 1005 X=EMTH 5(Y)                                                           MADA4810
      GO TO 1047                                                            MADA4820
 1006 X=EMTH 6(Y)                                                           MADA4830
      GO TO 1047                                                            MADA4840
 1007 X=EMTH 7(Y)                                                           MADA4850
      GO TO 1047                                                            MADA4860
 1008 X=EMTH 8(Y)                                                           MADA4870
      GO TO 1047                                                            MADA4880
 1009 X=EMTH 9(Y)                                                           MADA4890
      GO TO 1047                                                            MADA4900
 1010 X=EMTH10(Y)                                                           MADA4910
      GO TO 1047                                                            MADA4920
 1011 X=EMTH11(Y)                                                           MADA4930
      GO TO 1047                                                            MADA4940
 1012 X=EMTH12(Y)                                                           MADA4950
      GO TO 1047                                                            MADA4960
 1013 X=EMTH13(Y)                                                           MADA4970
      GO TO 1047                                                            MADA4980
 1014 X=EMTH14(Y)                                                           MADA4990
      GO TO 1047                                                            MADA5000
 1015 X=EMTH15(Y)                                                           MADA5010
 1047 IF(X.EQ.0.0)X=1.0E-6                                                  MADA5020
      TEMH(KGHS,I,J)=1.0/X                                                  MADA5030
 1050 CONTINUE                                                              MADA5040
CE    ***************************************************************************MADA5050
```

```
CE    ******************************************************************
 1299 RETURN
C=====================================================================
C/    CALCULATE THE CONDUCTIVITY OF THE GAS IN THE GAP
C=====================================================================
 1300 GK=(TH+TB)*0.5
      GO TO (1301,1302,1303,1304,1305,1306,1307,1308,1309,1310,1311,
     1       1312,1313,1314,1315),NI
 1301 GK=GCON1(X)
      GO TO 1350
 1302 GK=GCON2(X)
      GO TO 1350
 1303 GK=GCON3(X)
      GO TO 1350
 1304 GK=GCON4(X)
      GO TO 1350
 1305 GK=GCON5(X)
      GO TO 1350
 1306 GK=GCON6(X)
      GO TO 1350
 1307 GK=GCON7(X)
      GO TO 1350
 1308 GK=GCON8(X)
      GO TO 1350
 1309 GK=GCON9(X)
      GO TO 1350
 1310 GK=GCON10(X)
      GO TO 1350
 1311 GK=GCON11(X)
      GO TO 1350
 1312 GK=GCON12(X)
      GO TO 1350
 1313 GK=GCON13(X)
      GO TO 1350
 1314 GK=GCON14(X)
      GO TO 1350
 1315 GK=GCON15(X)
 1350 CONTINUE
CE    ******************************************************************
      RETURN
      END
```

```
      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 EMAH 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
```

```
      FUNCTION FMAT2  (X)
      ENTRY HEAT 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY RCON 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY ACON 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY TCON 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY SPEC 2(X)
      FMAT2 = 1.0
      GO TO 10
      ENTRY EMRL 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY EMRH 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY EMAL 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY EMAH 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY EMTL 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY EMTH 2(X)
      FMAT2 = 0.0
      GO TO 10
      ENTRY GCON 2(X)
      FMAT2 = 0.0
   10 RETURN
      END
```

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=====================================================================
C     SUPPLY THE COOLANT THERMAL PHYSICAL PROPERTIES
C=====================================================================
C=====================================================================
      DIMENSION        IFLODA(  4),STA    (  2)
      DATA (IFLODA(I),I=1,4)  /6HFLODAT, 1H1, 1H2, 1H3 /
      INTEGER          SELECT, TYPER, TYPEZ, TYPET
C=====================================================================
C=====================================================================
C=====================================================================
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=====================================================================
C     SUPPLY THE FLOW PROPERTIES
C=====================================================================
C     COOLANT NUMBER STORED AS NEGATIVE MATERIAL NUMBER
    1 N=IABS(MB(L))
C=====================================================================
C     IS THIS COOLANT IN USE? -'NO,YES,-
      IF(FLOW(N).EQ.0.0) RETURN
C=====================================================================
C     ASSIGN THE CURRENT TIME,INLET AND OUTLET TEMPERATURE
      HR = DATI
      TIN=TI(N)
      TOUT=TO(N)
C=====================================================================
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=====================================================================
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
      IF (X . LE . FLIM4(N)) GO TO 800
CE    **************************************************************************
C     THE INDEPENDENT VALUE LIES OUTSIDE ALL RANGES
   56 IERROR(1)=IFLODA(1)
      IERROR(2)=IFLODA(2)
      CALL ERROR2
C==========================================================================
C/    EVALUATE THE FLOWRATE USING THE LOW RANGE FLOW FUNCTION.
  600 GO TO(61,62,63,64,65,66,67,68,69,70,711,712,713,714,715),N
C==========================================================================
C     LOWER RANGE FLOW RATE FUNCTION
   61 FLOW(1) = FL01A(X)
      GO TO 900
   62 FLOW(2) = FL02A(X)
      GO TO 900
   63 FLOW(3) = FL03A(X)
      GO TO 900
   64 FLOW(4) = FL04A(X)
      GO TO 900
   65 FLOW(5) = FL05A(X)
      GO TO 900
   66 FLOW(6) = FL06A(X)
      GO TO 900
   67 FLOW(7) = FL07A(X)
      GO TO 900
   68 FLOW(8) = FL08A(X)
      GO TO 900
   69 FLOW(9) = FL09A(X)
      GO TO 900
   70 FLOW(10) = FL010A(X)
      GO TO 900
  711 FLOW(11)=FL011A(X)
      GO TO 900
  712 FLOW(12)=FL012A(X)
      GO TO 900
  713 FLOW(13)=FL013A(X)
      GO TO 900
  714 FLOW(14)=FL014A(X)
      GO TO 900
  715 FLOW(15)=FL015A(X)
      GO TO 900
CE    **************************************************************************
C/    EVALUATE THE FLOWRATE USING THE MIDDLE RANGE FLOW FUNCTION
  700 GO TO(71,72,73,74,75,76,77,78,79,80,811,812,813,814,815),N
   71 FLOW(1) = FL01B(X)
```

```
      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 TI(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(31,32,33,34,35,36,37,38,39,40,411,412,413,414,415),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
   39 TI(9) = TIN9B(X)
      GO TO 500
   40 TI(10) = TIN10B(X)
      GO TO 500
  411 TI(11)=TIN11B(X)
      GO TO 500
  412 TI(12)=TIN12B(X)
      GO TO 500
  413 TI(13)=TIN13B(X)
      GO TO 500
  414 TI(14)=TIN14B(X)
      GO TO 500
  415 TI(15)=TIN15B(X)
      GO TO 500
CE    *******************************************************************
C/    EVALUATE THE INLET TEMPERATURE USING THE HIGH RANGE FUNCTION
  400 GO TO(41,42,43,44,45,46,47,48,49,50,511,512,513,514,515),N
   41 TI(1) = TIN1C(X)
      GO TO 500
   42 TI(2) = TIN2C(X)
      GO TO 500
   43 TI(3) = TIN3C(X)
      GO TO 500
   44 TI(4) = TIN4C(X)
      GO TO 500
   45 TI(5) = TIN5C(X)
      GO TO 500
   46 TI(6) = TIN6C(X)
      GO TO 500
   47 TI(7) = TIN7C(X)
      GO TO 500
   48 TI(8) = TIN8C(X)
      GO TO 500
   49 TI(9) = TIN9C(X)
      GO TO 500
   50 TI(10) = TIN10C(X)
      GO TO 500
  511 TI(11)=TIN11C(X)
      GO TO 500
  512 TI(12)=TIN12C(X)
      GO TO 500
  513 TI(13)=TIN13C(X)
      GO TO 500
  514 TI(14)=TIN14C(X)
      GO TO 500
```

```
  515 TI(15)=TIN15C(X)
CE    *****************************************************************
C     =================================================================
C     CALCULATE HEAT TRANSFER COEFFICIENTS AND SPECIFIC HEATS FOR
C     THE COOLANTS.
C     =================================================================
C     SET ALL COOLANT CONDUCTIVITIES TO ZERO
  500 DO 510 K = KLS,KHS
      DO 510 J = JLS,JHS
      DO 510 I = ILS,IHS
      CONR(I,J,K)=1.E8
      CONT(I,J,K)=1.E8
      CONZ(I,J,K)=1.E8
  510 CONTINUE
      M=IABS(IPATH(N))
C     DETERMINE THE DIRECTION OF THE FLOW -'RADIAL,AXIAL,THETA'-
      GO TO(1000,2000,3000),M
C     *****************************************************************
CB    RADIAL COOLANT FLOW
C     *****************************************************************
C     DETERMINE THE BLOCK TYPE IN THE THETA DIRECTION.
 1000 CALL BLKTYP(KHS,KLS,KGHS,KGLS,NTG,TYPET)
C     DETERMINE THE BLOCK TYPE IN THE AXIAL DIRECTION.
      CALL BLKTYP(JHS,JLS,JGHS,JGLS,NZG,TYPEZ)
C     CALCULATE THE COOLANT PROPERTIES AT ALL LEVELS ALONG THE COOLANT
      DO 1500 I=ILS,IHS
C     SET THE LEVEL AND LOCAL COOLANT TEMPERATURE
      FTR=RP(I)
      DR=T(I,JLS,KLS)
C     CALCULATE THE SPECIFIC HEAT AT THIS LEVEL
      RCPC(N,I)=SPEC(X)
C     EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS
C     WHICH EXTEND OVER THE ENTIRE THETA DISTANCE OF THE PROBLEM.
      IF(KGHS.EQ.NTG.AND.KGLS.EQ.1) GO TO 1210
C     CALCULATE THE HEATTRANSFER COEFFICIENTS TO THE THETA WALLS
      DO 1200 J=JLS,JHS
      GO TO(1214,1212,1214,1216),TYPET
C     CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH
C     THETA BOUNDARIES OF THE COOLANT BLOCK.
 1212 STA(1)=TBBTH(KGLS,I,J)
      STA(2)=TBBTL(KGHS,I,J)
      CONT(I,J,KLS)=1./HC((STA(1)+STA(2))*.5)
      GO TO 1200
C     CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE
C     THETA BOUNDARIES OF THE COOLANT BLOCK.
 1214 STA(2)=TBBTL(KGHS,I,J)
```

```
      CONT(I,J,KHS)=1./HC(STA(2))
      IF(TYPET.NE.1) GO TO 1200
 1216 STA(1)=TBBTH(KGLS,I,J)
      CONT(I,J,KLS)=1./HC(STA(1))
C     HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED?
C    .-'YES,NO'-
 1200 CONTINUE
C     EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS
C     WHICH EXTEND OVER THE ENTIRE AXIAL DISTANCE OF THE PROBLEM.
 1210 IF(JGHS.EQ.NZG.AND.JGLS.EQ.1) GO TO 1500
C     CALCULATE THE HEATTRANSFER COEFFICIENTS TO THE AXIAL WALLS
      DO 1301 K=KLS,KHS
      GO TO(1224,1222,1224,1226),TYPEZ
C     CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH
C     AXIAL BOUNDARIES OF THE COOLANT BLOCK.
 1222 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 1301
C     CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE
C     AXIAL BOUNDARIES OF THE COOLANT BLOCK.
 1224 STA(2)=ZBBTL(JGHS,I,K)
      CONZ(I,JHS,K)=1./HC(STA(2))
      IF(TYPEZ.NE.1) GO TO 1301
 1226 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'-
 1301 CONTINUE
C     HAVE THE COOLANT PROPERTIES BEEN CALCULATED AT ALL LEVELS?
C    .-'YES,NO'-
 1500 CONTINUE
CE    ******************************************************************
      RETURN
C     ******************************************************************
CB    AXIAL COOLANT FLOW
C     ******************************************************************
C     DETERMINE THE BLOCK TYPE IN THE THETA DIRECTION.
 2000 CALL BLKTYP(KHS,KLS,KGHS,KGLS,NTG,TYPET)
C     DETERMINE THE BLOCK TYPE IN THE RADIAL DIRECTION.
      CALL BLKTYP(IHS,ILS,IGHS,IGLS,NRG,TYPER)
C     CALCULATE THE COOLANT PROPERTIES AT ALL LEVELS ALONG THE COOLANT
      DO 2500 J=JLS,JHS
C     SET THE LEVEL AND LOCAL COOLANT TEMPERATURE
      FTZ=ZP(J)
      DR=T(ILS,J,KLS)
```

```
C     CALCULATE THE SPECIFIC HEAT AT THIS LEVEL
      RCPL(N,J)=SPEC(X)
C     EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS
C     WHICH EXTEND OVER THE ENTIRE THETA DISTANCE OF THE PROBLEM.
      IF(KGHS.EQ.NTG.AND.KGLS.EQ.1) GO TO 2210
C     CALCULATE THE HEATTRANSFER COEFFICIENTS TO THE THETA WALLS
      DO 2200 I=ILS,IHS
      GO TO(2214,2212,2214,2216),TYPET
C     CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH
C     THETA BOUNDARIES OF THE COOLANT BLOCK.
 2212 STA(1)=TBBTH(KGLS,I,J)
      STA(2)=TBBTL(KGHS,I,J)
      CONT(I,J,KLS)=1./HC((STA(1)+STA(2))*.5)
      GO TO 2200
C     CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE
C     THETA BOUNDARIES OF THE COOLANT BLOCK.
 2214 STA(2)=TBBTL(KGHS,I,J)
      CONT(I,J,KHS)=1./HC(STA(2))
      IF(TYPET.NE.1) GO TO 2200
 2216 STA(1)=TBBTH(KGLS,I,J)
      CONT(I,J,KLS)=1./HC(STA(1))
C     HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED?
C     .-'YES,NO'-
 2200 CONTINUE
C     EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS
C     WHICH EXTEND OVER THE ENTIRE RADIAL DISTANCE OF THE PROBLEM.
 2210 IF(IGHS.EQ.NRG.AND.IGLS.EQ.1) GO TO 2500
C     CALCULATE THE HEATTRANSFER COEFFICIENTS TO THE RADIAL WALLS
      DO 2300 K=KLS,KHS
      GO TO(2224,2222,2224,2226),TYPER
C     CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH
C     RADIAL BOUNDARIES OF THE COOLANT BLOCK.
 2222 STA(1)=RBBTH(IGLS,J,K)
      STA(2)=RBBTL(IGHS,J,K)
      CONR(ILS,J,K)=1./HC((STA(1)+STA(2))*.5)
      GO TO 2300
C     CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR THE
C     RADIAL BOUNDARIES OF THE COOLANT BLOCK.
 2224 STA(2)=RBBTL(IGHS,J,K)
      CONR(IHS,J,K)=1./HC(STA(2))
      IF(TYPER.NE.1) GO TO 2300
 2226 STA(1)=RBBTH(IGLS,J,K)
      CONR(ILS,J,K)=1./HC(STA(1))
C     HAVE ALL HEAT TRANSFER COEFFICIENTS AT THIS LEVEL BEEN CALCULATED?
C     .-'YES,NO'-
 2300 CONTINUE
```

```
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    THETA 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
      DR=T(ILS,JLS,K)                                                    FLOD4300
C     CALCULATE THE SPECIFIC HEAT AT THIS LEVEL                          FLOD4310
      RCPC(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
      CONR(ILS,J,K)=1./HC((STA(1)+STA(2))*.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
      CONR(IHS,J,K)=1./HC(STA(2))                                        FLOD4480
      IF(TYPER.NE.1) GO TO 3200                                          FLOD4490
 3216 STA(1)=RBBTH(IGLS,J,K)                                             FLOD4500
      CONR(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
```

```
 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,    FLOD5280
     *1420,1430,1440,1450),N                                           FLOD5290
C======================================================================FLOD5300
 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)
      GO TO (1331,1332,1333),IA
 1331 B=H3A(X)
      GO TO 1399
 1332 B=H3B(X)
      GO TO 1399
 1333 B=H3C(X)
      GO TO 1399
C=======================================================================
 1340 RE=KEYN4(X)
      IA=SELECT(N)
      GO TO (1341,1342,1343),IA
 1341 B=H4A(X)
      GO TO 1399
 1342 B=H4B(X)
      GO TO 1399
 1343 B=H4C(X)
      GO TO 1399
C=======================================================================
 1350 RE=KEYN5(X)
      IA=SELECT(N)
      GO TO (1351,1352,1353),IA
 1351 B=H5A(X)
      GO TO 1399
 1352 B=H5B(X)
      GO TO 1399
 1353 B=H5C(X)
      GO TO 1399
C=======================================================================
 1360 RE=KEYN6(X)
      IA=SELECT(N)
      GO TO (1361,1362,1363),IA
 1361 B=H6A(X)
      GO TO 1399
 1362 B=H6B(X)
      GO TO 1399
 1363 B=H6C(X)
      GO TO 1399
C=======================================================================
 1370 RE=KEYN7(X)
      IA=SELECT(N)
      GO TO (1371,1372,1373),IA
 1371 B=H7A(X)
      GO TO 1399
 1372 B=H7B(X)
      GO TO 1399
```

```
 1373 B=H7C(X)                                                                FLOD5980
      GO TO 1399                                                              FLOD5990
C=============================================================================FLOD6000
 1380 RE=REYN8(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=============================================================================FLOD6100
 1390 RE=REYN9(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=============================================================================FLOD6200
 1400 RE=REYN10(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=============================================================================FLOD6300
 1410 RE=REYN11(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=============================================================================FLOD6400
 1420 RE=REYN12(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=====================================================================FLOD6500
 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=====================================================================FLOD6600
 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=====================================================================FLOD6700
 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=====================================================================FLOD6790
 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
      INTEGER FUNCTION SELECT(N)                                      FLOD6870
C=====================================================================FLOD6880
C       DOES THE REYNOLDS NUMBER LIE WITHIN THE SPECIFIED RANGE?      FLOD6890
```

```
C     . -'NO,YES'-
      IF(RE.LT.RLIM1(N).OR.RE.GT.RLIM4(N))GO TO 500
C     DOES THE REYNOLDS NUMBER LIE IN THE LOW RANGE? -'YES,NO'-
      IF(RE.LT.RLIM2(N))GO TO 100
C     DOES THE REYNOLDS NUMBER LIE IN THE MIDDLE RANGE? -'YES,NO'-
      IF(RE.LT.RLIM3(N))GO TO 200
C     UPPER RANGE HEAT TRANSFER CORRELATION
      SELECT=3
      RETURN
C     LOWER RANGE HEAT TRANSFER ROUTINE
  100 SELECT=1
      RETURN
C     MIDDLE RANGE HEAT TRANSFER CORRELATION
  200 SELECT=2
      RETURN
C======================================================================
C     THE REYNOLDS NUMBER LIES OUTSIDE THE SPECIFIED RANGE
  500 IERROR(1)=IFLODA(1)
      IERROR(2)=IFLODA(4)
      CALL ERROR2
C======================================================================
      END
```

```
      SUBROUTINE BLKTYP(XHS,XLS,XGHS,XGLS,NXG,TYPE)
C     ======================================================================
C     DETERMINE COOLANT BLOCK TYPE WITH RESPECT TO BOUNDARIES ALONG
C     WHICH HEAT TRANSFER COEFFICIENTS ARE TO BE ASSIGNED.
C        TYPE=1   ASSIGN SEPARATE VALUES ALONG BOTH BOUNDARIES.
C        TYPE=2   ASSIGN AN AVERAGE OF THE SEPARATE VALUES ALONG
C                 BOTH BOUNDARIES.
C        TYPE=3   ASSIGN A VALUE ALONG THE HIGH BOUNDARY ONLY.
C        TYPE=4   ASSIGN A VALUE ALONG THE LOW BOUNDARY ONLY.
C     ======================================================================
      INTEGER          XHS, XLS, XGHS, XGLS, TYPE
C     ======================================================================
C     ======================================================================
      TYPE=0
C     IS THIS AN INTERNAL COOLANT BLOCK ?
      IF(XGHS.NE.XGLS) GO TO 100
C     SHOULD THE HEAT TRANSFER COEFFICIENT BE CALCULATED FOR THE HIGH
C     OR THE LOW BOUNDARY OF THE COOLANT BLOCK ?
      IF(XGHS.EQ.1) GO TO 110
      GO TO 120
C     IS THE INTERNAL COOLANT BLOCK ADJACENT TO AN EXTERNAL COOLANT ON
C     EITHER OF THE TWO BOUNDARIES BEING CONSIDERED ?
  100 IF(XGHS.EQ.NXG) GO TO 120
      IF(XGLS.EQ.1) GO TO 110
C     IS THE COOLANT BLOCK ONLY ONE POINT WIDE IN THE DIRECTION
C     BEING CONSIDERED ?
      IF(XLS.NE.XHS) GO TO 110
      TYPE=2
      GO TO 130
  110 TYPE=3
      IF(XGLS.EQ.1) GO TO 130
      TYPE=1
      GO TO 130
  120 TYPE=4
  130 RETURN
      END
```

```
      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
```

```
      FUNCTION COOL2  (X)
      ENTRY SPH2(X)
      COOL2 = 1.0
      GO TO 10
      ENTRY REYN2(X)
      COOL2 = 0.0
      GO TO 10
      ENTRY H2A(X)
      COOL2 = 0.0
      GO TO 10
      ENTRY H2B(X)
      COOL2 = 0.0
      GO TO 10
      ENTRY H2C(X)
      COOL2 = 0.0
      GO TO 10
      ENTRY FLO2A(X)
      COOL2 = 1000000.0
      GO TO 10
      ENTRY FLO2B(X)
      COOL2 = 0.0
      GO TO 10
      ENTRY FLO2C(X)
      COOL2 = 0.0
      GO TO 10
      ENTRY TIN2A(X)
      COOL2 = 460.0
      GO TO 10
      ENTRY TIN2B(X)
      COOL2 = 0.0
      GO TO 10
      ENTRY TIN2C(X)
      COOL2 = 0.0
   10 RETURN
      END
```

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 CONDUC
      INCLUDE       COMDIM
C     ==================================================================
CB    CALCULATE THE AVERAGE CONDUCTIVITIES BETWEEN POINTS. RADIATION
C     ACROSS GAPS AND INTERNAL COOLANTS IS INCLUDED.
C     ==================================================================
C     ERROR MESSAGES
C        CONDUC1 , CONDUC2 , CONDUC3
C                        THESE MESSAGES INDICATE THAT RADIATION BETWEEN THE
C                        BOUNDARIES OF AN INTERNAL COOLANT BLOCK WAS NEGLECTED
C                        IN ORDER TO COMPLETE AN ITERATION. A SMALLER TIME
C                        STEP SHOULD BE USED IF THE NEGLECT OF RADIATION IS
C                        SIGNIFICANT.
C
C          CONDUC1  RADIATION WAS NEGLECTED BETWEEN RADIAL GAPLINES IGL
C                   AND IGH AT AXIAL POINT LEVEL J,THETA POINT LEVEL K.
C          CONDUC2  RADIATION WAS NEGLECTED BETWEEN AXIAL GAPLINES JGL
C                   AND JGH AT RADIAL POINT LEVEL I,THETA POINT LEVEL K.
C          CONDUC3  RADIATION WAS NEGLECTED BETWEEN THETA GAPLINES KGL
C                   AND KGH AT RADIAL POINT LEVEL I,AXIAL POINT LEVEL J.
C     ==================================================================
      LOGICAL           NOMES
      DIMENSION         ICOND(13)
      DATA(ICOND(I),I=1,13)/ 1H1,5H IGL=,5H IGH=,3H J=,3H K=,1H2,
     *5H JGL=,5H JGH=,3H I=,1H3,5H KGL=,5H KGH=,3H K=/
  100 FORMAT(16H ERROR AT CONDUC,A1,17H ON ITERATION NO.,I4,3H AT,A5,I2,
     *A5,I2,A3,I2,A3,I2)
C     ==================================================================
C     SET THE CORRECT ENTRY FOR MADATA
      GAS = .TRUE.
C     ==================================================================
      TDMIN = 0.10
      NNITER = NITER + 1
      NOMES=.FALSE.
      IF(.NOT.SW(10)) GO TO 105
      IF(CS1.LT.1.0) NOMES=.TRUE.
  105 IF(FIRST) NOMES=.TRUE.
C     ==================================================================
C     CALCULATE THE AVERAGE RADIAL CONDUCTIVITIES BETWEEN POINTS.
C=    ==================================================================
C     CALCULATE THE RADIAL CONDUCTIVITIES IN EACH RADIAL PLANE.
      DO 340 I=1,IM
      IG = IGR(I)
C=    ==================================================================
C     CALCULATE THE RADIAL CONDUCTIVITIES IN EACH AXIAL COLUMN IN THE
```

```
C      PLANE.
       DO 330 J=2,JM
C      ==================================================================
C      CALCULATE THE RADIAL CONDUCTIVITIES IN THE COLUMN.
       DO 320 K=2,KM
C      DOES THE GRIDLINE HAVE ANYWHERE A GAP OR A COOLANT ADJACENT TO IT
C     .? -'NO,YES'-
       IF (IG.EQ.0) GO TO 110
C      ==================================================================
C      IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS
C      POINT ? -'NO,YES'-
       IF (GAPR(IG,J,K).EQ.0.0) GO TO 110
C      IS IT A GAP OR A COOLANT WHICH IS ADJACENT ? -'COOLANT,GAP'-
       IF (GAPR(IG,J,K).LT.0.0) GO TO 130
C      ==================================================================
CB     CALCULATE THE AVERAGE CONDUCTIVITY WITH A GAP PRESENT
       NI = MATRG(IG, J, K)
       TH = RBBTH(IG, J, K)
       TB = RBBTL(IG, J, K)
       FTR = RP(I)
       FTZ = ZP(J)
       FTT=TP(K)
C      CALCULATE THE LOCAL GAS CONDUCTIVITY.
       CALL MADATA
       BX = REMH(IG, J, K) + REML(IG, J, K) - 1.0
       CX = .1713E-8*(TH**2 + TB**2)*(TH + TB)/BX
       X1 = CONR(I, J, K)*RATIOB(IG, J, K)
       X2 = 1./((GK/GAPR(IG, J, K) + CX)*RATIOH(IG))
       X3 = CONR(I + 1, J, K)
C      PRESERVE THE DATA REQUIRED FOR CALCULATING RADIAL GAP BOUNDARY
C     .TEMPERATURES BY STORING IT IN THE REM-RBBT LOCATIONS FOR THE
C     .CURRENT GAPLINE.
       REML(IG, J, K) = CONR(I, J, K)
       REMH(IG, J, K) = CONR(I + 1, J, K)
       RBBTL(IG, J, K) = BX
       RBBTH(IG, J, K) = GK
       KR(I, J, K) = 1./((X1 + X2 + X3)/RATIOK(IG))
       GO TO 310
CE     ******************************************************************
CE     ******************************************************************
C      ARE THE CONDUCTIVITIES ON BOTH SIDES OF THE GRID PLANE IDENTICAL
C     .? -'NO,YES'-
  110  IF (CONR(I,J,K).NE.CONR(I+1,J,K)) GO TO 120
C      ASSIGN ONE OF THEM AS THE AVERAGE CONDUCTIVITY
       KR(I, J, K) = 1.0/CONR(I, J, K)
       GO TO 310
```

```
C     CALCULATE THE AVERAGE CONDUCTIVITY WHEN NO GAP OR COOLANT IS
C     PRESENT
  120 KR(I, J, K) = 1.0/(CONR(I + 1, J, K) + (CONR(I, J, K) - CONR(I + 1
     *  , J, K))*RLN(I))
      GO TO 310
CB    CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ADJACENT
C     IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GRIDPLANE?-*NO,YES*-
  130 IF (GAPR(IG,J,K).LT.(-1.5E-10)) GO TO 210
C     CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE HIGH
C     INDEX SIDE.
C     EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.
      IF (IG.EQ.NRG) GO TO 190
C     FIND THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      IG1 = IG + 1
      DO 150 II=IG1,NRG
      IF (GAPR(II,J,K).LT.(-1.5E-10)) GO TO 140
      GO TO 150
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT THE
C     HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  140 IF (GAPR(II,J,K).LT.(-2.5E-10)) GO TO 190
      IGH = II
      GO TO 160
  150 CONTINUE
  160 IF (ABS(REMH(IG,J,K)).GT..999E6.OR.ABS(REML(IGH,J,K)).GT..999E6) G
     1O TO 170
      IF (ABS(T(I,J,K)-T(I+1,J,K)).LT.TDMIN) GO TO 180
      RAD = 0.1713E-8*(RBBTH(IG, J, K)**4 - RBBTL(IGH, J, K)**4)/(REMH(
     * IG, J, K) + REML(IGH, J, K) - 1.0)
      HX = CONR(I + 1, J, K)*(T(I, J, K) - T(I + 1, J, K) - RAD*CONR(I,
     * J, K)*DELR(I)*RLN(I))/(T(I, J, K) - T(I + 1, J, K) + RAD*CONR(I
     *  + 1, J, K))
      MATRG(IG, J, K) = 200
      MATRG(IGH, J, K) = 200
      GO TO 200
  170 MATRG(IG, J, K) = 100
      MATRG(IGH, J, K) = 100
      GO TO 190
  180 MATRG(IG, J, K) = 300
      MATRG(IGH, J, K) = 300
  190 HX = CONR(I + 1, J, K)
  200 KR(I, J, K) = 1.0/(CONR(I, J, K)*RLN(I) + HX*RATIOC(IG))
      GO TO 310
C     ARE TWO COOLANTS ADJACENT? -*NO,YES*-
  210 IF (GAPR(IG,J,K).GT.(-2.5E-10)) GO TO 220
C     DO NOT ALLOW HEAT TRANSFER BETWEEN COOLANTS
      KR(I, J, K) = 1.0E-10
```

```
      GO TO 310
C     CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE
C     LOW INDEX SIDE.
C     EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.
  220 IF (IG.EQ.1) GO TO 290
C     FIND THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      IG1 = IG - 1
      DO 240 II=1,IG1
      IG2 = IG - II
      IF (GAPR(IG2,J,K).LT.(-.5E-10)) GO TO 230
      GO TO 240
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT THE
C     LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  230 IF (GAPR(IG2,J,K).LT.(-2.5E-10)) GO TO 290
      IGL = IG2
      GO TO 250
  240 CONTINUE
  250 IF (MATRG(IG,J,K).EQ.100.OR.MATRG(IG,J,K).EQ.300) GO TO 290
      IF (ABS(T(I,J,K)-T(I+1,J,K)).LT.TDMIN) GO TO 260
      RAD = 0.1713E-8*(RBBTH(IGL, J, K)**4 - RBBTL(IG, J, K)**4)/(REMH(
     * IGL, J, K) + REML(IG, J, K) - 1.0)
      HX = CONR(I, J, K)*(T(I + 1, J, K) - T(I, J, K) + CONR(I + 1, J, K
     * )*RAD+RATIOB(IG, J, K)*DELR(I)/RATIOK(IG))/(T(I + 1, J, K) - T(I
     * , J, K) - CONR(I, J, K)*RAD*RATIOB(IG, J, K))
      GO TO 300
  260 MATRG(IG, J, K) = 300
      MATRG(IGL, J, K) = 300
      DO 270 IX=1,IM
      IF (IGR(IX).NE.IGL) GO TO 270
      IPL = IX
      GO TO 280
  270 CONTINUE
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING AND PRESERVED IN THE REM-RBBT ARRAYS.
C     CONL IS CONR(IPL,J,K)
  280 CONL=REML(IGL,J,K)
C     CONH IS CONR(IPL+1,J,K)
      CONH = RBBTL(IGL, J, K)
      HX=CONH
      KR(IPL, J, K) = 1.0/(CONL*RLN(IPL) + HX*RATIOC(IGL))
  290 HX = CONR(I, J, K)
  300 KR(I, J, K) = 1.0/(CONR(I + 1, J, K)/RATIOK(IG) + HX*RATIOC(IG))
      IF (MATRG(IG,J,K).NE.300) GO TO 310
      IF(NOMES) GO TO 310
      IERROR(1) = IERROR(1) + 1
      WRITE (6,100) ICOND(1),NNITER,ICOND(2),IGL,ICOND(3),IG,ICOND(4),J,
```

```
     1ICOND(13),K
CE    **********************************************************************
  310 CONTINUE
C     HAVE ALL POINTS IN THE COLUMN BEEN CONSIDERED ? -'YES,NO'-
  320 CONTINUE
C     HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'-
  330 CONTINUE
C     HAVE ALL PLANES BEEN CONSIDERED ? -'YES,NO'-
  340 CONTINUE
CE    **********************************************************************
C     ======================================================================
C     ======================================================================
C     CALCULATE THE AVERAGE AXIAL CONDUCTIVITIES BETWEEN POINTS.
C     ======================================================================
C     CALCULATE THE AXIAL CONDUCTIVITIES IN EACH AXIAL PLANE.
      DO 580 J=1,JM
      JG = JGZ(J)
C     ======================================================================
C     CALCULATE THE AXIAL CONDUCTIVITIES IN EACH RADIAL COLUMN IN THE
C     PLANE.
      DO 570 I=2,IM
C     ======================================================================
C     CALCULATE THE AXIAL CONDUCTIVITIES IN THE COLUMN.
      DO 560 K=2,KM
C     DOES THE GRIDLINE HAVE ANYWHERE A GAP OR A COOLANT ADJACENT TO IT
C    .? -'NO,YES'-
      IF (JG.EQ.0) GO TO 350
C     ======================================================================
C     IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS
C     POINT ? -'NO,YES'-
      IF (GAPZ(JG,I,K).EQ.0.0) GO TO 350
C     IS IT A GAP OR A COOLANT WHICH IS ADJACENT ? -'COOLANT,GAP'-
      IF (GAPZ(JG,I,K).LT.0.0) GO TO 370
C     ======================================================================
CB    CALCULATE THE AVERAGE CONDUCTIVITY WITH A GAP PRESENT
      NI = MATZG(JG, I, K)
      TH = ZGBTH(JG, I, K)
      TB = ZGBTL(JG, I, K)
      FTR = RP(I)
      FTZ = ZP(J)
      FTT=TP(K)
C     CALCULATE THE LOCAL GAS CONDUCTIVITY.
      CALL MADATA
      BX = ZEMH(JG, I, K) + ZEML(JG, I, K) - 1.0
      CX = .1713E-8*(TH**2 + TB**2)*(TH + TB)/BX
      X1 = CONZ(I, J, K)*ZATIOB(JG, I, K)
```

```
      X2=1./((GK/GAPZ(JG,I,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-ZBBT LOCATIONS FOR THE      COND2330
C    .CURRENT GAPLINE.                                                  COND2340
      ZEML(JG, I, K) = CONZ(I, J, K)                                    COND2350
      ZEMH(JG, I, K) = CONZ(I, J + 1, K)                                COND2360
      ZBBTL(JG, I, K) = GX                                              COND2370
      ZBBTH(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) - CONZ(I, J COND2510
     ,  + 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) - ZP(J)                                               COND2590
C     EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.                 COND2600
      IF (JG.EQ.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(ZEML(JGH,I,K)).GT..999E6) GCOND2730
     1O 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(
     * JG, I, K) + ZEML(JGH, I, K) - 1.0)
      HX = CONZ(I, J + 1, K)*(T(I, J, K) - T(I, J + 1, K) - RAD*CONZ(I,
     * J, K)*DXT)/(T(I, J, K) - T(I, J + 1, K) + RAD*CONZ(I, J + 1, K))
      MATZG(JG, I, K) = 200
      MATZG(JGH, I,K)=200
      GO TO 440
  410 MATZG(JG, I, K) = 100
      MATZG(JGH, I,K)=100
      GO TO 430
  420 MATZG(JG, I, K) = 300
      MATZG(JGH, I,K)=300
  430 HX = CONZ(I, J + 1, K)
  440 KZ(I,J,K)=DELZ(J)/(CONZ(I,J,K)*DXT+HX)
      GO TO 550
C     ARE TWO COOLANTS ADJACENT? -,NO,YES,-
  450 IF (GAPZ(JG,I,K).GT.(-2.5E-10)) GO TO 460
C     DO NOT ALLOW HEAT TRANSFER BETWEEN COOLANTS
      KZ(I, J, K) = 1.0E-10
      GO TO 550
C     CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE LOW
C     INDEX SIDE.
  460 DXT = ZH(J + 1) - ZL(J)
C     EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.
      IF (JG.EQ.1) GO TO 530
C     FIND THE LOW AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      JG1 = JG - 1
      DO 480 JJ=1,JG1
      JG2 = JG - JJ
      IF (GAPZ(JG2,I,K).LT.(-.5E-10)) GO TO 470
      GO TO 480
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C     THE LOW AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  470 IF (GAPZ(JG2,I,K).LT.(-2.5E-10)) GO TO 530
      JGL = JG2
      GO TO 490
  480 CONTINUE
  490 IF (MATZG(JG,I,K).EQ.100.OR.MATZG(JG,I,K).EQ.300) GO TO 530
      IF (ABS(T(I,J,K)-T(I,J+1,K)).LT.TDMIN) GO TO 500
      RAD = 0.1713E-8*(ZBBTH(JGL, I, K)**4 - ZBBTL(JG, I, K)**4)/(ZEMH(
     * JGL, I, K) + ZEML(JG, I, K) - 1.0)
      HX = CONZ(I, J, K)*(T(I, J + 1, K) - T(I, J, K) + CONZ(I, J + 1, K
     * )*RAD*DXT)/(T(I, J + 1, K) - T(I, J, K) - CONZ(I, J, K)*RAD)
      GO TO 540
  500 MATZG(JG, I, K) = 300
      MATZG(JGL, I, K) = 300
```

```
      DO 510 JX=1,JM
      IF (JGZ(JX).NE.JGL) GO TO 510
      JPL = JX
      GO TO 520
  510 CONTINUE
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE ZEM-ZBBT ARRAYS.
C     CONL IS CONZ(I,JPL,K)
  520 CONL=ZEML(JGL,I,K)
C     CONH IS CONZ(I,JPL+1,K)
      CONH=ZHBTL(JGL,I,K)
      HX=CONH
      DXTP=(ZL(JPL)-ZP(JPL))
      KZ(I,JPL,K)=DELZ(JPL)/(CONL*DXTP+HX)
  530 HX = CONZ(I, J, K)
  540 KZ(I,J,K)=DELZ(J)/(CONZ(I,J+1,K)*DXT+HX)
      IF (MATZG(JG,I,K).NE.300) GO TO 550
      IF(NOMES) GO TO 550
      IERROR(1) = IERROR(1) + 1
      WRITE (6,100) ICOND(6),NNITER,ICOND(7),JGL,ICOND(8),JG,ICOND(9),I,
     1ICOND(13),K
CE    ******************************************************************
  550 CONTINUE
C     HAVE ALL POINTS IN THE COLUMN BEEN CONSIDERED ? -'YES,NO'-
  560 CONTINUE
C     HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'-
  570 CONTINUE
C     HAVE ALL PLANES BEEN CONSIDERED ? -'YES,NO'-
  580 CONTINUE
CE    ******************************************************************
C     ==================================================================
C     ==================================================================
C     CALCULATE THE AVERAGE THETA CONDUCTIVITIES BETWEEN POINTS.
C     ==================================================================
C     CALCULATE THE THETA CONDUCTIVITIES IN EACH THETA PLANE.
      DO 640 K=1,KM
      KG=KGT(K)
C     ==================================================================
C     CALCULATE THE THETA CONDUCTIVITIES IN EACH AXIAL COLUMN IN THE
C     PLANE.
      DO 630 I=2,IM
C     ==================================================================
C     CALCULATE THE THETA CONDUCTIVITIES IN THE COLUMN.
      DO 620 J=2,JM
C     DOES THE GRIDLINE HAVE ANYWHERE A GAP OR A COOLANT ADJACENT TO IT
C    ',? -'NO,YES'-
```

```
      IF (KG.EQ.0) GO TO 600                                            COND3680
C     ==================================================================COND3690
C     IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS      COND3700
C     POINT ? -'NO,YES'-                                                COND3710
      IF (GAPT(KG,I,J).EQ.0.0) GO TO 600                                COND3720
C     IS IT A GAP OR A COOLANT WHICH IS ADJACENT ? -'COOLANT,GAP'-      COND3730
      IF (GAPT(KG,I,J).LT.0.0) GO TO 620                                COND3740
C     ==================================================================COND3750
CB    CALCULATE THE AVERAGE CONDUCTIVITY WITH A GAP PRESENT             COND3760
      NI = MATTG(KG, I, J)                                              COND3770
      TH = TBBTH(KG, I, J)                                              COND3780
      TB = TBBTL(KG, I, J)                                              COND3790
      FTR = RP(I)                                                       COND3800
      FTZ = ZP(J)                                                       COND3810
      FTT=TP(K)                                                         COND3820
      GWID = GAPT(KG, I, J)                                             COND3830
      DXHS=TATIOH(KG)                                                   COND3840
C     CONVERT GWID AND DXHS FROM RADIANS TO FEET FOR CYLINDRICAL        COND3850
C     GEOMETRY.                                                         COND3860
      IF (ISHAPE.NE.0) GO TO 590                                        COND3870
       GWID = GWID*RP(I)                                                COND3880
      DXHS=DXHS*RP(I)                                                   COND3890
C     CALCULATE THE LOCAL GAS CONDUCTIVITY.                             COND3900
  590 CALL MADATA                                                       COND3910
      BX = TEMH(KG, I, J) + TEML(KG, I, J) - 1.0                        COND3920
      CX = .1713E-8*(TH**2 + TB**2)*(TH + TB)/BX                        COND3930
      X1 = CONT(I, J, K)*TATIOB(KG, I, J)                               COND3940
      X2=1./((GK/GWID+CX)*DXHS)                                         COND3950
      X3 = CONT(I, J, K + 1)                                            COND3960
C     PRESERVE THE DATA REQUIRED FOR CALCULATING THETA GAP BOUNDARY     COND3970
C     TEMPERATURES BY STORING IT IN THE TEM-TBBT LOCATIONS FOR THE      COND3980
C     CURRENT GAPLINE.                                                  COND3990
      TEML(KG, I, J) = CONT(I, J, K)                                    COND4000
      TEMH(KG, I, J) = CONT(I, J, K + 1)                                COND4010
      TBBTL(KG, I, J) = BX                                              COND4020
      TBBTH(KG, I, J) = GK                                              COND4030
      KT(I,J,K)=1./((X1+X2+X3)/TATIOK(KG))                              COND4040
      GO TO 810                                                         COND4050
CE    ******************************************************************COND4060
CE    ******************************************************************COND4070
C     ARE THE CONDUCTIVITIES ON BOTH SIDES OF THE GRID PLANE IDENTICAL  COND4080
C    .? -'NO,YES'-                                                      COND4090
  600 IF (CONT(I,J,K).NE.CONT(I,J,K+1)) GO TO 610                       COND4100
C     ASSIGN ONE OF THEM AS THE AVERAGE CONDUCTIVITY                    COND4110
      KT(I, J, K) = 1.0/CONT(I, J, K)                                   COND4120
      GO TO 810                                                         COND4130
```

```
C       CALCULATE THE AVERAGE CONDUCTIVITY WHEN NO GAP OR COOLANT IS
C       PRESENT
  610 KT(I, J, K) = 1.0/(CONT(I, J, K + 1) + (CONT(I, J, K) - CONT(I, J
     ., K+1))*TLN(K))
      GO TO 810
CB      CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ADJACENT
  620 DZ=DELT(K)
C       CONVERT DZ FROM RADIANS TO FEET FOR CYLINDRICAL GEOMETRY.
      IF(ISHAPE.EQ.0) DZ=DZ*RP(I)
C       IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GRIDPLANE?-,NO,YES'.
      IF (GAPT(KG,I,J).LT.(-1.5E-10)) GO TO 700
C       CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE HIGH
C       INDEX SIDE.
      DXT=TL(K)-TP(K)
C       CONVERT DXT FROM RADIANS TO FEET FOR CYLINDRICAL GEOMETRY.
      IF(ISHAPE.EQ.0) DXT=DXT*RP(I)
C       EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.
      IF(KG.EQ.NTG) GO TO 680
C       FIND THE HIGH THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      KG1=KG+1
      DO 640 KK=KG1,NTG
      IF (GAPT(KK,I,J).LT.(-1.5E-10)) GO TO 630
      GO TO 640
C       EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C       THE HIGH THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  630 IF (GAPT(KK,I,J).LT.(-2.5E-10)) GO TO 680
      KGH=KK
      GO TO 650
  640 CONTINUE
  650 IF (ABS(TEMH(KG,I,J)).GT..999E6.OR.ABS(TEML(KGH,I,J)).GT..999E6) G
     1O TO 660
      IF (ABS(T(I,J,K)-T(I,J,K+1)).LT.TDMIN) GO TO 670
      RAD = 0.1713E-8*(TBBTH(KG, I, J)**4 - TBBTL(KGH, I, J)**4)/(TEMH(
     , KG, I, J) + TEML(KGH, I, J) - 1.0)
      HX = CONT(I, J, K + 1)*(T(I, J, K) - T(I, J, K + 1) - RAD*CONT(I,
     , J, K)*DXT)/(T(I, J, K) - T(I, J, K + 1) + RAD*CONT(I, J, K + 1))
      MATIG(KG, I, J) = 200
      MATIG(KGH, I, J) = 200
      GO TO 690
  660 MATIG(KG, I, J) = 100
      MATIG(KGH, I, J) = 100
      GO TO 680
  670 MATIG(KG, I, J) = 300
      MATIG(KGH, I, J) = 300
  680 HX = CONT(I, J, K + 1)
  690 KT(I, J, K) = DZ/(CONT(I, J, K)*DXT + HX)
```

```
      GO TO 810
C     ARE TWO COOLANTS ADJACENT? -,NO,YES,-
  700 IF (GAPT(KG,I,J).GT.(-2.5E-10)) GO TO 710
C     DO NOT ALLOW HEAT TRANSFER BETWEEN COOLANTS
      KT(I, J, K) = 1.0E-10
      GO TO 810
C     CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE LOW
C     INDEX SIDE.
  710 DXT=TP(K+1)-TL(K)
C     CONVERT DXT FROM RADIANS TO FEET FOR CYLINDRICAL GEOMETRY.
      IF(ISHAPE.EQ.0) DXT=DXT*RP(I)
C     EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.
      IF(KG.EQ.1) GO TO 790
C     FIND THE LOW THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      KG1=KG-1
      DO 730 KK=1,KG1
      KG2=KG-KK
      IF (GAPT(KG2,I,J).LT.(-.5E-10)) GO TO 720
      GO TO 730
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C     THE LOW THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  720 IF (GAPT(KG2,I,J).LT.(-2.5E-10)) GO TO 790
      KGL=KG2
      GO TO 740
  730 CONTINUE
  740 IF (MATTG(KG,I,J).EQ.100.OR.MATTG(KG,I,J).EQ.300) GO TO 790
      IF (ABS(T(I,J,K)-T(I,J,K+1)).LT.TDMIN) GO TO 750
      RAD = 0.1713E-8*(TBBTH(KGL, I, J)**4 - TBBTL(KG, I, J)**4)/(TEMH(
     * KGL, I, J) + TEML(KG, I, J) - 1.0)
      HX = CONT(I, J, K)*(T(I, J, K + 1) - T(I, J, K) + CONT(I, J, K + 1
     * )*RAD*DXT)/(T(I, J, K + 1) - T(I, J, K) - CONT(I, J, K)*RAD)
      GO TO 800
  750 MATTG(KG, I, J) = 300
      MATTG(KGL, I, J) = 300
      DO 760 KX=1,KM
      IF (KGT(KX).NE.KGL) GO TO 760
      KPL=KX
      GO TO 770
  760 CONTINUE
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE TEM-TBBT ARRAYS.
C     CONL IS CONT(I,J,KPL)
  770 CONL=TEML(KGL,I,J)
C     CONH IS CONT(I,J,KPL+1)
      CONH = TBBTL(KGL, I, J)
      HX=CONH
```

```
      DZP=DELT(KPL)
      DXTP= TL(KPL)-TP(KPL)
C     CONVERT DZP AND DXTP FROM RADIANS TO FEET FOR CYLINDRICAL
C     GEOMETRY.
      IF (ISHAPE.NE.0) GO TO 780
      DZP=DZP*RP(I)
      DXTP=DXTP*RP(I)
  780 KT(I,J,KPL)=DZP/(CONL*DXTP+HX)
  790 HX = CONT(I, J, K)
  800 KT(I, J, K) = DZ/(CONT(I, J, K + 1)*DXT + HX)
      IF (MATTG(KG,I,J).NE.300) GO TO 810
      IF(NOMES) GO TO 810
      IERROR(1) = IERROR(1) + 1
      WRITE (6,100) ICOND(10),NNITER,ICOND(11),JGL,ICOND(12),JG,ICOND(9)
     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
```

```
      SUBROUTINE STEP($,NOTEST)
      INCLUDE       COMDIM
C=====================================================================
CB    CALCULATE THE NEW TEMPERATURES AFTER ONE TIMESTEP
C=====================================================================
C=====================================================================
      LOGICAL           NOTEST
      DIMENSION         X       ( MQ),Y       ( MQ)
C=====================================================================
C/    PERFORM A RADIAL SET OF ITERATIONS
      DO 12 J= 2,JM
      DO 11 K= 2,KM
C=====================================================================
      I=2
      C1=KR(I-1,J,K)*KR(I-1,J,K)
      C7=KCP(I,J,K)/DT
      C2=KR(I,J,K)*KR(I,J,K)/2.0
      E=-C7
      X(I)=C2/E
      Y(I)=(-(C7*T(I,J,K)+C1*(T(I-1,J,K)-T(I,J,K)             1+C2*
     X(T(I+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)
     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)
     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)
     X-T(I,J,K))+W(I,J,K))             )/E
C=====================================================================
      DO 10 I=3,IM1
      C1=KR(I-1,J,K)*KR(I-1,J,K)/2.0
      C7=KCP(I,J,K)/DT
      C2=KR(I,J,K)*KR(I,J,K)/2.0
      E=-C7-C1*X(I-1)
      X(I)=C2/E
      Y(I)=(-(C7*T(I,J,K)+C1*(T(I-1,J,K)-T(I,J,K)-T(I,J,K))+C2*
     X(T(I+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)
     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)
     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)
     X-T(I,J,K))+W(I,J,K))-C1*Y(I-1))/E
   10 CONTINUE
C=====================================================================
      I= IM
      C1=KR(I-1,J,K)*KR(I-1,J,K)/2.0
      C7=KCP(I,J,K)/DT
      C2=KR(I,J,K)*KR(I,J,K)
      E=-C7-C1*X(I-1)
      X(I)=0.0
      Y(I)=(-(C7*T(I,J,K)+C1*(T(I-1,J,K)-T(I,J,K)-T(I,J,K))+C2*
```

```
     X(T(I+1,J,K)-T(I,J,K)          )+RZ(I,J-1,K)*KZ(I,J-1,K)*(T(I,J-1,K)
     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)
     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)
     X-T(I,J,K))+W(I,J,K))-C1*Y(I-1))/E
C=====================================================================
   15 TT(I,J,K)=Y(I)-X(I)*TT(I+1,J,K)
      I=I-1
      IF (I.GT.1) GO TO 15
C=====================================================================
   11 CONTINUE
   12 CONTINUE
CE    *****************************************************************
C=====================================================================
C/    PERFORM A AXIAL SET OF ITERATIONS
      DO 22 I= 2,IM
      DO 21 K= 2,KM
C=====================================================================
      DO 20 J=2,JM
      C3=RZ(I,J-1,K)*KZ(I,J-1,K)
      IF (J .EQ. 2)  C3 = 0.0
      C7=RCP(I,J,K)/DT*2.0
      C4=RZ(I,J,K)*KZ(I,J,K)
      IF (J .EQ. JM) C4 = 0.0
      E=-C7-C3*X(J-1)
      X(J)=C4/E
      Y(J)=(-C7*TT(I,J,K)+C3*T(I,J-1,K)+C4*T(I,J+1,K)-C3*Y(J-1))/E
   20 CONTINUE
C=====================================================================
      J= JM
   25 TT(I,J,K)=Y(J)-X(J)*TT(I,J+1,K)
      J=J-1
      IF (J.GT.1) GO TO 25
C=====================================================================
   21 CONTINUE
   22 CONTINUE
CE
C=====================================================================
C/    PERFORM A THETA SET OF ITERATIONS
      DO 32 I= 2,IM
      DO 31 J= 2,JM
C=====================================================================
      DO 30 K=2,KM
      C5=RT(I,J,K-1)*KT(I,J,K-1)
      IF (K .EQ. 2) C5 = 0.0
      C7=RCP(I,J,K)/DT*2.0
      C6=RT(I,J,K)*KT(I,J,K)
```

```
      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                                                             STEP1000
      IF (K.GT.1) GO TO 35                                              STEP1010
C=======================================================================STEP1020
   31 CONTINUE                                                          STEP1030
   32 CONTINUE                                                          STEP1040
CE    ******************************************************************STEP1050
      IF(.NOT.SW(10).OR.NOTEST) GO TO 50                                STEP1060
      DO 40 I=2,IM                                                      STEP1070
      DO 40 J=2,JM                                                      STEP1080
      DO 40 K=2,KM                                                      STEP1090
      IF(T(I,J,K).LT.0.0.OR.T(I,J,K).GT.1.0E6) RETURN 1                 STEP1100
   40 CONTINUE                                                          STEP1110
   50 RETURN                                                            STEP1120
C=======================================================================STEP1130
      END                                                               STEP1140
```

```
      SUBROUTINE COOL
      INCLUDE      COMDIM
C     ==================================================================
C     CALCULATE THE COOLANT TEMPERATURES
C     ==================================================================
C     ERROR MESSAGES
C        COOL1 , COOL2 , COOL3
C                THERE IS A PROGRAM LIMITATION WHICH MAY BE ENCOUNTERED
C                WHEN CALCULATING COOLANT TEMPERATURES IN THE PRESENCE
C                OF RADIATION BETWEEN COOLANT BLOCK BOUNDARIES. IT
C                ARISES WHEN A COOLANT POINT AND AN ADJACENT MATERIAL
C                POINT APPROACH THE SAME TEMPERATURE. THESE MESSAGES
C                INDICATE THAT THIS LIMITATION HAS BEEN ENCOUNTERED
C                AND CIRCUMVENTED BY ASSIGNING ZERO COOLANT HEAT
C                TRANSFER AT THE LEVEL INDICATED. BY USING A SMALLER
C                TIME STEP,THIS DIFFICULTY CAN USUALLY BE ELIMINATED.
C
C                COOL1   HEAT TRANSFER WITHIN A RADIAL FLOW COOLANT IS
C                        ZERO IN BLOCK L AT RADIAL POINT LEVEL I.
C                COOL2   HEAT TRANSFER WITHIN AN AXIAL FLOW COOLANT IS
C                        ZERO IN BLOCK L AT AXIAL POINT LEVEL J.
C                COOL3 . HEAT TRANSFER WITHIN A THETA FLOW COOLANT IS
C                        ZERO IN BLOCK L AT THETA POINT LEVEL K.
C     ==================================================================
      LOGICAL          ANTI,    NOMES,    IFP,    IRAD,    IXRAD
      DIMENSION        TC(MQ),  ICOOL(6)
C     ==================================================================
      DATA(ICOOL(I),I=1,6) / 1H1,1H2,1H3,3H I=,3H J=,3H K= /
  100 FORMAT(14H ERROR AT COOL,A1,17H ON ITERATION NO.,I4,6H AT L=,
     *I3,A3,I2)
C     ==================================================================
C     ==================================================================
      NNITER = NITER + 1
      NOMES=.FALSE.
      IF(.NOT.SW(10)) GO TO 105
      IF(CS1.LT.1.0) NOMES=.TRUE.
  105 IF(FIRST) NOMES=.TRUE.
C     ==================================================================
C     CALCULATE THE COOLANT TEMPERATURES IN EACH BLOCK.
      DO 650 L=1,LMAX
C     IS IT A COOLANT BLOCK ? -'NO,YES,-
      IF (MB(L).GE.0) GO TO 650
      N = - MB(L)
      TIN = TI(N)
C     DEFINE THE CORNERS OF THE COOLANT BLOCK
```

```
      ILS = IL(L)
      JLS = JL(L)
      KLS=KL(L)
      IHS = IH(L)
      JHS = JH(L)
      KHS=KH(L)
C     FIND THE HIGH BOUNDING GAPLINES OF THE COOLANT BLOCK.
      IF (IHS.EQ.IMAX) GO TO 110
      IGHS = IGR(IHS)
      GO TO 120
  110 IGHS = NRG
  120 IF (JHS.EQ.JMAX) GO TO 130
      JGHS = JGZ(JHS)
      GO TO 140
  130 JGHS = NZG
  140 IF (KHS.EQ.KMAX) GO TO 150
       KGHS=KGT(KHS)
      GO TO 160
  150 KGHS=NTG
C     DETERMINE THE DIRECTION OF COOLANT FLOW -'RADIAL,AXIAL,THETA'-
  160 ANTI = .FALSE.
      IF(IPATH(N).LT.0) ANTI=.TRUE.
      I = IABS(IPATH(N))
      GO TO (490,170,330), I
C     ==================================================================
C     CALCULATE THE AXIAL COOLANT TEMPERATURES.
C     ==================================================================
C     ASSIGN THE INLET TEMPERATURE
  170 TC(JLS)=TIN
      IF(ANTI) TC(JHS)=TIN
C     CALCULATE THE COOLANT TEMPERATURES ALONG THE FLOW DIRECTION.
      DO 320 J1=JLS,JHS
      J=J1
      IF(ANTI) J=JHS+JLS-J
      SUMKA = 0.0
      SUMKAT = 0.0
      IFP = .FALSE.
      IXRAD = .FALSE.
C     CALCULATE THE HEAT TRANSFER ACROSS THE COOLANT BOUNDARIES AT THE
C     CURRENT AXIAL LEVEL.
C     SUM THE HEAT TRANSFER ACROSS THE RADIAL BOUNDARIES OF THE COOLANT.
      DO 190 K=KLS,KHS
C     IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX
C    .SIDE? -'YES,NO'-
      IF (ILS.LE.2) GO TO 180
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 = RR(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'-,
  180 IF (IHS.GE.IM) GO TO 190
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 = RR(I, J, K)*KR(I, J, K)
      SUMKAT = SUMKAT + X*T(I + 1, J, K)
      SUMKA = SUMKA + X
      IF(IRAD) CALL RANGE(IFP,T(I+1,J,K),THI,TLO)
C     HAS THE HEAT TRANSFER THROUGH ALL RADIAL POINTS BEEN
C    .CALCULATED? -'YES,NO'-
  190 CONTINUE
C     SUM THE HEAT TRANSFER ACROSS THE THETA BOUNDARIES OF THE COOLANT.
      DO 210 I=ILS,IHS
C     IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX
C    .SIDE? -'YES,NO'-
      IF (KLS.LE.2) GO TO 200
C     SUM THE HEAT TRANSFER ACROSS THE LOW THETA BOUNDARY.
      K=KLS-1
      IRAD = .FALSE.
      IF(MATTG(KGHS,I,J).EQ.200) IRAD=.TRUE.
      IF(IRAD) IXRAD=.TRUE.
      X = RT(I, J, K)*KT(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'-
  200 IF (KHS.GE.KM) GO TO 210
C     SUM THE HEAT TRANSFER ACROSS THE HIGH THETA BOUNDARY.
      K=KHS
      IRAD = .FALSE.
      IF(MATTG(KGHS,I,J).EQ.200) IRAD=.TRUE.
      IF(IRAD) IXRAD=.TRUE.
      X = RT(I, J, K)*KT(I, J, K)
```

```
      SUMKAT = SUMKAT + X*T(I, J, K + 1)
      SUMKA = SUMKA + X
      IF(IRAD) CALL RANGE(IFP,T(I,J,K+1),THI,TLO)
C     HAS THE HEAT TRANSFER THROUGH ALL THETA POINTS BEEN
C     CALCULATED ? -'YES,NO'-
  210 CONTINUE
C     CALCULATE THE COOLANT TEMPERATURES FOR THE CURRENT AXIAL
C     INCREMENT.
      WCP = FLOW(N)*RCPC(N, J)
      EX = SUMKA/WCP
C     IS THE RADIATION EFFECT UPON THE HEAT TRANSFER COEFFICIENT TOO
C    .LARGE? -'NO,YES'-
      IF (EX.GT.-30.) GO TO 220
      GO TO 270
  220 EXPO = EXP( - EX)
C     CALCULATE THE OUTLET COOLANT TEMPERATURE FOR THE AXIAL INCREMENT.
      X = TC(J)*EXPO + SUMKAT/SUMKA*(1.0 - EXPO)
C     CALCULATE THE AVERAGE COOLANT TEMPERATURE FOR THE AXIAL INCREMENT.
  230 TAV = (SUMKAT - WCP*(X - TC(J)))/SUMKA
C     DOES THE MEAN TEMPERATURE FALL BETWEEN THE INLET AND EXIT
C     TEMPERATURES ?
      IF (TC(J).GT.X) GO TO 240
      IF (TAV.GE.TC(J).AND.TAV.LE.X) GO TO 260
      GO TO 250
  240 IF (TAV.GE.X.AND.TAV.LE.TC(J)) GO TO 260
  250 TAV = (TC(J) + X)*0.5
C     DOES THE EXIT TEMPERATURE FALL WITHIN THE LIMITS IMPOSED BY THE
C     INLET TEMPERATURE AND THE ADJACENT MATERIAL TEMPERATURES ?
  260 IF (.NOT.IXRAD) GO TO 280
      IF(TC(J).GT.THI) THI=TC(J)
      IF(TC(J).LT.TLO) TLO=TC(J)
      IF (X.LE.THI.AND.X.GE.TLO) GO TO 280
  270 X = TC(J)
      TAV = TC(J)
      IF(NOMES) GO TO 280
      IERROR(1) = IERROR(1) + 1
      WRITE (6,100) ICOOL(2),NNITER,L,ICOOL(5),J
  280 IF (ANTI) GO TO 290
      TC(J+1)=X
      GO TO 300
  290 TC(J-1)=X
C     RECORD THE COOLANT TEMPERATURE THROUGHOUT THE CURRENT
C     AXIAL INCREMENT.
  300 DO 310 I=ILS,IHS
      DO 310 K=KLS,KHS
      T(I, J, K) = TAV
```

```
  310 CONTINUE
C     HAVE ALL AXIAL LEVELS BEEN CALCULATED ? -'YES,NO'-
  320 CONTINUE
C     ASSIGN THE COOLANT OUTLET TEMPERATURE.
      TO(N) = X
      GO TO 650
C     ==================================================================
C     CALCULATE THE THETA COOLANT TEMPERATURES.
C     ==================================================================
C     ASSIGN THE INLET TEMPERATURE
  330 TC(KLS)=TIN
      IF(ANTI) TC(KHS)=TIN
C     CALCULATE THE COOLANT TEMPERATURES ALONG THE FLOW DIRECTION.
      DO 480 K1=KLS,KHS
      K=K1
      IF(ANTI) 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 = RR(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.IM) 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 = RR(I, J, K)*KR(I, J, K)
      SUMKAT = SUMKAT + X*T(I + 1, J, K)
```

```
      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.JI) 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 INCREMENTCOOL2640
      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 (EX.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
C     TEMPERATURES ?
      IF (TC(K).GT.X) GO TO 400
      IF (TAV.GE.TC(K).AND.TAV.LE.X) GO TO 420
      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
C     INLET TEMPERATURE AND THE ADJACENT MATERIAL TEMPERATURES ?
  420 IF (.NOT.IXRAD) GO TO 440
      IF(TC(K).GT.THI) THI=TC(K)
      IF(TC(K).LT.TLO) TLO=TC(K)
      IF (X.LE.THI.AND.X.GE.TLO) GO TO 440
  430 X = TC(K)
      TAV = TC(K)
      IF(NOMES) GO TO 440
      IERROR(1) = IERROR(1) + 1
      WRITE (6,100) ICOOL(3),NNITER,L,ICOOL(6),K
  440 IF (ANTI) GO TO 450
      TC(K+1)=X
      GO TO 460
  450 TC(K-1)=X
C     RECORD THE COOLANT TEMPERATURE THROUGHOUT THE CURRENT
C     THETA INCREMENT.
  460 DO 470 I=ILS,IHS
      DO 470 J=JLS,JHS
      T(I, J, K) = TAV
  470 CONTINUE
C     HAVE ALL THETA LEVELS BEEN CALCULATED ? -'YES,NO'-
  480 CONTINUE
C     ASSIGN THE COOLANT OUTLET TEMPERATURE.
      TO(N) = X
      GO TO 650
C     ==================================================================
C     CALCULATE THE RADIAL COOLANT TEMPERATURES.
C     ==================================================================
C     ASSIGN THE INLET TEMPERATURE
  490 TC(ILS)=TIN
      IF(ANTI) TC(IHS)=TIN
C     CALCULATE THE COOLANT TEMPERATURES ALONG THE FLOW DIRECTION.
      DO 640 I1=ILS,IHS
      I=I1
      IF(ANTI) I=IHS+ILS-I
      SUMKA = 0.0
      SUMKAT = 0.0
      IFP = .FALSE.
```

```
      IXRAD = .FALSE.
C     CALCULATE THE HEAT TRANSFER ACROSS THE COOLANT BOUNDARIES AT THE
C     CURRENT RADIAL LEVEL.
C     SUM THE HEAT TRANSFER ACROSS THE AXIAL BOUNDARIES OF THE COOLANT.
      DO 510 K=KLS,KHS
C     IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX
C    .SIDE? -'YES,NO'-
      IF (KLS.LE.2) GO TO 500
C     SUM THE HEAT TRANSFER ACROSS THE LOW AXIAL BOUNDARY.
      J=JLS-1
      IRAD = .FALSE.
      IF(MATZG(JGHS,I,K).EQ.200) IRAD=.TRUE.
      IF(IRAD) IXRAD=.TRUE.
      X = RZ(I, J, K)*KZ(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'-
  500 IF (JHS.GE.JM) GO TO 510
C     SUM THE HEAT TRANSFER ACROSS THE HIGH AXIAL BOUNDARY.
      J=JHS
      IRAD = .FALSE.
      IF(MATZG(JGHS,I,K).EQ.200) IRAD=.TRUE.
      IF(IRAD) IXRAD=.TRUE.
      X = RZ(I, J, K)*KZ(I, J, K)
      SUMKAT = SUMKAT + X*T(I, J + 1, K)
      SUMKA = SUMKA + X
      IF(IRAD) CALL RANGE(IFP,T(I,J+1,K),THI,TLO)
C     HAS THE HEAT TRANSFER THROUGH ALL AXIAL POINTS BEEN
C    .CALCULATED? -'YES,NO'-
  510 CONTINUE
C     SUM THE HEAT TRANSFER ACROSS THE THETA BOUNDARIES OF THE COOLANT.
      DO 530 J=JLS,JHS
C     IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX
C    .SIDE? -'YES,NO'-
      IF (KLS.LE.2) GO TO 520
C     SUM THE HEAT TRANSFER ACROSS THE LOW THETA BOUNDARY.
      K=KLS-1
      IRAD = .FALSE.
      IF(MATTG(KGHS,I,J).EQ.200) IRAD=.TRUE.
      IF(IRAD) IXRAD=.TRUE.
      X = RT(I, J, K)*KT(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          COOL3680
C     .SIDE? -'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(MATTG(KGHS,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*T(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 INCREMENTCOOL3940
  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 (6,190) ICOOL(3),NNITER,L,ICOOL(4),I                        COOL4130
```

```
  600 IF (ANTI) GO TO 610
      TC(I+1)=X
      GO TO 620
  610 TC(I-1)=X
C     RECORD THE COOLANT TEMPERATURE THROUGHOUT THE CURRENT
C     RADIAL INCREMENT.
  620 DO 630 J=JLS,JHS
      DO 630 K=KLS,KHS
      T(I, J, K) = TAV
  630 CONTINUE
C     HAVE ALL RADIAL LEVELS BEEN CALCULATED ? -'YES,NO'-
  640 CONTINUE
C     ASSIGN THE COOLANT OUTLET TEMPERATURE.
      TO(N) = X
C     ====================================================================
C     HAVE ALL BLOCKS BEEN CHECKED    -'YES,NO'-
  650 CONTINUE
      RETURN
C     ====================================================================
C     ********************************************************************
C     ********************************************************************
C     ********************************************************************
C     ====================================================================
      SUBROUTINE RANGE(IFP, TX, THI, TLO)
C     ====================================================================
C     DETERMINE THE CURRENT VALUES OF THE HIGHEST AND LOWEST ADJACENT
C     MATERIAL POINT TEMPERATURES.
C     ====================================================================
      LOGICAL IFP
C     ====================================================================
      IF (IFP) GO TO 100
      THI = TX
      TLO = TX
      IFP = .TRUE.
      GO TO 110
  100 IF(TX.GT.THI) THI=TX
      IF(TX.LT.TLO) TLO=TX
C     ====================================================================
  110 RETURN
      END
```

```
      SUBROUTINE SURT
      INCLUDE       COMD1M
C=    ==================================================================
C     CALCULATE THE BOUNDARY TEMPERATURES ASSOCIATED WITH THE POINT
C    .TEMPERATURE RESULTS OF THE CURRENT ITERATION.
C=    ==================================================================
C     ERROR STOPS=
C        SURT  1   INSTABILITY ENCOUNTERED WHILE CALCULATING THE
C                 RADIAL BOUNDARY TEMPERATURES OF A GAP. USE A
C                  SMALLER TIME STEP.
C        SURT  2   INSTABILITY ENCOUNTERED WHILE CALCULATING THE
C                 AXIAL BOUNDARY TEMPERATURES  OF A GAP. USE A
C                  USE A SMALLER TIME STEP.
C       SURT  3   INSTABILITY ENCOUNTERED WHILE CALCULATING THE
C                 THETA BOUNDARY TEMPERATURES OF A GAP. USE A
C                 SMALLER TIME STEP.
C     ==================================================================
      DIMENSION       ISURT(4)
      DATA(ISURT(I),I=1,4)  /6HSURT  , 1H1, 1H2, 1H3/
C     ==================================================================
C     CALCULATE THE BOUNDARY TEMPERATURES ADJACENT GAPS AND COOLANTS.
C     ==================================================================
C     CALCULATE THE RADIAL BOUNDARY TEMPERATURES IN EACH RADIAL PLANE.
      DO 280 I=1,IM
      IG=IGR(I)
C     EXCLUDE THE NON-GAPLINES.
      IF (IG.EQ.0) GO TO 280
C     ==================================================================
C     CALCULATE THE RADIAL BOUNDARY TEMPERATURES FOR EACH AXIAL
C     COLUMN IN THE PLANE.
      DO 270 J=2,JM
C     ==================================================================
C     CALCULATE THE RADIAL BOUNDARY TEMPERATURES IN THE COLUMN.
      DO 260 K=2,KM
C     IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS
C    .POINT ? -'NO,YES'-
      IF (GAPR(IG,J,K).EQ.0.0) GO TO 260
C     IS IT A COOLANT OR A GAP WHICH IS ADJACENT ? -'COOLANT,GAP'-
      IF (GAPR(IG,J,K).LT.0.0) GO TO 130
C     ==================================================================
C     CALCULATE THE BOUNDARY TEMPERATURES WHEN A GAP IS ADJACENT.
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE REM-RBBT ARRAYS.
C     BX IS (REMH(IG,J,K)+REML(IG,J,K)-1.0).
      BX = RBBTL(IG, J, K)
```

```
C     GK IS THE GAS THE GAS THERMAL CONDUCTIVITY.
      GK = RBBTH(IG, J, K)
C     CONL IS CONR(I,J,K)
      CONL = REML(IG, J, K)
C     CONH IS CONR(I+1,J,K)
      CONH = REMH(IG, J, K)
C     GET READY TO CALCULATE THE BOUNDARY TEMPERATURES OF THE GAP.
      B = RATIOB(IG, J, K)*CONL/CONH
      C = T(I, J, K) + B*T(I + 1, J, K)
      RGAP = 0.1713E-8*GAPR(IG, J, K)/RX
      ED = RATIOH(IG)*CONH/GAPR(IG, J, K)
C     ALLOW 10 ITERATIONS TO CALCULATE THE BOUNDARY TEMPERATURES.
      DO 120 KK=1,10
      IF (KK.EQ.1) GO TO 100
      TH = RBBTH(IG, J, K)
      TB = RBBTL(IG, J, K)
      GO TO 110
  100 TH = T(I, J, K)
      TB = T(I + 1, J, K)
  110 GH=(GK+RGAP*(TH**2+TB**2)*(TH+TB))*ED
      X1 = GH*C
      X2 = 1.0 + GH*(B + 1.0)
      RBBTL(IG, J, K) = (X1 + T(I + 1, J, K))/X2
      RBBTH(IG, J, K) = (X1 + T(I, J, K))/X2
C     ARE THE BOUNDARY TEMPERATURES CONVERGED ? -'YES,NO'-
      IF (ABS(RBBTH(IG,J,K)-TH).LE.1.0) GO TO 260
C     HAVE THE 10 ITERATIONS BEEN PERFORMED ? -'YES,NO'-
  120 CONTINUE
C     NEGATIVE TEMPERATURES ARE PRESENT AND RESULT IN LOCAL
C    .INSTABILITIES. TRY A SMALLER TIME STEP.
      IERROR(1) = ISURT(1)
      IERROR(2) = ISURT(2)
      CALL ERROR2
C     ================================================================
C     CALCULATE THE BOUNDARY TEMPERATURES WHEN A COOLANT IS ADJACENT.
C     IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE ? -'NO,YES'-
  130 IF (GAPR(IG,J,K).LT.(-1.5E-10)) GO TO 190
C     CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE HIGH
C    .INDEX SIDE.
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE REM-RBBT ARRAYS.
C     CONL IS CONR(I,J,K)
      CONL = REML(IG, J, K)
C     CONH IS CONR(I+1,J,K)
      CONH = RBBTL(IG, J, K)
C     EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.
```

```
      IF (IG.EQ.NRG.OR.MATRG(IG,J,K).NE.200) GO TO 170
C     FIND THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      IG1 = IG + 1
      DO 150 II=IG1,NRG
      IF (GAPR(II,J,K).LT.(-1.5E-10)) GO TO 140
      GO TO 150
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C     THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  140 IF (GAPR(II,J,K).LT.(-2.5E-10)) GO TO 170
      IGH = II
      GO TO 160
  150 CONTINUE
  160 RAD = 0.1713E-8*(RBBTH(IG, J, K)**4 - RBBTL(IGH, J, K)**4)/(REMH(
     * IG, J, K) + REML(IGH, J, K) - 1.0)
C     PRESERVE RAD IN THE REML ARRAY.
      REML(IG, J, K) = RAD
      HX = CONH*(T(I, J, K) - T(I + 1, J, K) - RAD*CONL*DELR(I)*RLN(I))/
     * (T(I, J, K) - T(I + 1, J, K) + RAD*CONH)
      GO TO 180
  170 HX = CONH
C     CALCULATE NEW BOUNDARY TEMPERATURES.
  180 RBBTH(IG, J, K) = (CONL*T(I + 1, J, K) + HX/(DELR(I)*RLN(I))*T(I,
     * J, K))/(CONL + HX/(DELR(I)*RLN(I)))
      RBBTL(IG, J, K) = RBBTH(IG, J, K)
      GO TO 260
C     ARE TWO COOLANTS ADJACENT ? -'NO,YES'-
  190 IF (GAPR(IG,J,K).GT.(-2.5E-10)) GO TO 200
C     DO NOT ATTEMPT TO CALCULATE BOUNDARY TEMPERATURES.
      GO TO 260
C     CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE LOW
C    .INDEX SIDE.
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE REM-RBBT ARRAYS.
C     CONL IS CONR(I,J,K)
  200 CONL=RBBTH(IG,J,K)
C     CONH IS CONR(I+1,J,K)
      CONH=REMH(IG,J,K)
C     EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.
      IF (IG.EQ.1) GO TO 240
      IF (MATRG(IG,J,K).NE.200) GO TO 240
C     FIND THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      IG1 = IG - 1
      DO 220 II=1,IG1
      IG2 = IG - II
      IF (GAPR(IG2,J,K).LT.(-.5E-10)) GO TO 210
      GO TO 220
```

```
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C     THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  210 IF (GAPR(IG2,J,K).LT.(-2.5E-10)) GO TO 240
      IGL = IG2
      GO TO 230
  220 CONTINUE
C     RECOVER VALUE OF RAD STORED IN THE REML ARRAY.
  230 RAD = REML(IGL, J, K)
      HX = CONL*(T(I + 1, J, K) - T(I, J, K) + CONH*RAD*RATIOB(IG, J, K)
     ' *DELR(I)/RATIOK(IG))/(T(I + 1, J, K) - T(I, J, K) - CONL*RAD*
     ' RATIOB(IG, J, K))
      GO TO 250
  240 HX=CONL
C     CALCULATE NEW BOUNDARY TEMPERATURES.
  250 RBBTL(IG, J, K) = (CONH*T(I, J, K) + HX*RATIOK(IG)/DELR(I)*T(I + 1
     ' , J, K))/(CONH + HX*RATIOK(IG)/DELR(I))
      RBBTH(IG, J, K) = RBBTL(IG, J, K)
C     ==================================================================
C     HAVE ALL POINTS IN THE COLUMN BEEN CONSIDERED ? -'YES,NO'-
  260 CONTINUE
C     ==================================================================
C     HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'-
  270 CONTINUE
C     ==================================================================
C     HAVE ALL RADIAL BOUNDARY TEMPERATURES BEEN CALCULATED ?-'YES,NO'-
  280 CONTINUE
C     ==================================================================
C     ==================================================================
C     CALCULATE THE AXIAL BOUNDARY TEMPERATURES IN EACH AXIAL PLANE.
      DO 470 J=1,JM
      JG = JGZ(J)
C     ==================================================================
C     EXCLUDE THE NON-GAPLINES.
      IF (JG.EQ.0) GO TO 470
C     ==================================================================
C     CALCULATE THE AXIAL BOUNDARY TEMPERATURES FOR EACH RADIAL COLUMN
C     COLUMN IN THE PLANE.
      DO 460 I=2,IM
C     ==================================================================
C     CALCULATE THE AXIAL BOUNDARY TEMPERATURES IN THE COLUMN.
      DO 450 K=2,KM
C     IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS
C    .POINT ? -'NO,YES'-
      IF (GAPZ(JG,I,K).EQ.0.0) GO TO 450
C     IS IT A COOLANT OR A GAP WHICH IS ADJACENT ? -'COOLANT,GAP'-
      IF (GAPZ(JG,I,K).LT.0.0) GO TO 320
```

```
C     ==================================================================
C     CALCULATE THE BOUNDARY TEMPERATURES WHEN A GAP IS ADJACENT.
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE ZEM-ZBBT ARRAYS.
C     BX IS (ZEMH(JG,I,K)+ZEML(JG,I,K)-1.0)
      BX = ZBBTL(JG, I, K)
C     GK IS THE GAS THERMAL CONDUCTIVITY.
      GK = ZBBTH(JG, I, K)
C     CONL IS CONZ(I,J,K)
      CONL = ZEML(JG, I, K)
C     CONH IS CONZ(I,J+1,K)
      CONH = ZEMH(JG, I, K)
C     GET READY TO CALCULATE THE BOUNDARY TEMPERATURES OF THE GAP.
      B = ZATIOB(JG, I, K)*CONL/CONH
      C = T(I, J, K) + B*T(I, J + 1, K)
      RGAP = 0.1713E-8*GAPZ(JG, I, K)/BX
      ED = ZATIOH(JG)*CONH/GAPZ(JG, I, K)
C     ALLOW 10 ITERATIONS TO CALCULATE THE BOUNDARY TEMPERATURES.
      DO 310 KK=1,10
      IF (KK.EQ.1) GO TO 290
      TH = ZBBTH(JG, I, K)
      TB = ZBBTL(JG, I, K)
      GO TO 300
  290 TH = T(I, J, K)
      TB = T(I, J + 1, K)
  300 GH = (GK + RGAP*(TH**2 + TB**2)*(TH + TB))*ED
      X1 = GH*C
      X2 = 1.0 + GH*(B + 1.0)
      ZBBTL(JG, I, K) = (X1 + T(I, J + 1, K))/X2
      ZBBTH(JG, I, K) = (X1 + T(I, J, K))/X2
C     ARE THE BOUNDARY TEMPERATURES CONVERGED ? -'YES,NO'-
      IF (ABS(ZBBTH(JG,I,K)-TH).LE.1.0) GO TO 450
C     HAVE THE 10 ITERATIONS BEEN PERFORMED ? -'YES,NO'-
  310 CONTINUE
C     NEGATIVE TEMPERATURES ARE PRESENT AND RESULT IN LOCAL
C    .INSTABILITIES. TRY A SMALLER TIME STEP.
      IERROR(1) = ISURT(1)
      IERROR(2) = ISURT(3)
      CALL ERROR2
C     ==================================================================
C     CALCULATE THE BOUNDARY TEMPERATURES WHEN A COOLANT IS ADJACENT.
C     IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE ? -'NO,YES'-
  320 IF (GAPZ(JG,I,K).LT.(-1.5E-10)) GO TO 380
C     CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE HIGH
C    .INDEX SIDE.
      BXT = ZL(J) - ZP(J)
```

```
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE ZEM-ZBBT ARRAYS.
C     CONL IS CONZ(I,J,K)
      CONL = ZEML(JG, I, K)
C     CONH IS CONZ(I,J+1,K)
      CONH = ZBBTL(JG, I, K)
C     EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.
      IF (JG.EQ.NZG.OR.MATZG(JG,I,K).NE.200) GO TO 360
C     FIND THE HIGH AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      JG1 = JG + 1
      DO 340 JJ=JG1,NZG
      IF (GAPZ(JJ,I,K).LT.(-1.5E-10)) GO TO 330
      GO TO 340
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C     THE HIGH AXIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  330 IF (GAPZ(JJ,I,K).LT.(-2.5E-10)) GO TO 360
      JGH = JJ
      GO TO 350
  340 CONTINUE
  350 RAD = 0.1713E-8*(ZBBTH(JG, I, K)**4 - ZBBTL(JGH, I, K)**4)/(ZEMH(
     * JG, I, K) + ZEML(JGH, I, K) - 1.0)
C     PRESERVE RAD IN THE ZEML ARRAY.
      ZEML(JG, I, K) = RAD
      HX = CONH*(T(I, J, K) - T(I, J + 1, K) - RAD*CONL*DXT)/(T(I, J, K)
     *  - T(I, J + 1, K) + RAD*CONH)
      GO TO 370
  360 HX = CONH
C     CALCULATE NEW BOUNDARY TEMPERATURES.
  370 ZBBTH(JG, I, K) = (CONL*T(I, J + 1, K) + HX/DXT*T(I, J, K))/(CONL
     *  + HX/DXT)
      ZBBTL(JG, I, K) = ZBBTH(JG, I, K)
      GO TO 450
C     ARE TWO COOLANTS ADJACENT ? -'NO,YES'-
  380 IF (GAPZ(JG,I,K).GT.(-2.5E-10)) GO TO 390
C     DO NOT ATTEMPT TO CALCULATE BOUNDARY TEMPERATURES.
      GO TO 450
C     CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE LOW .
C    .INDEX SIDE.
  390 DXT = ZP(J + 1) - ZL(J)
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE ZEM-ZBBT ARRAYS.
C     CONL IS CONZ(I,J,K)
      CONL = ZBBTH(JG, I, K)
C     CONH IS CONZ(I,J+1,K)
      CONH = ZEMH(JG, I, K)
C     EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.
```

```
      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 ZBBTL(JG, I, K) = (CONH*T(I, J, K) + HX/DXT*T(I, J + 1, K))/(CONH
     *  + HX/DXT)
      ZBBTH(JG, I, K) = ZBBTL(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     =================================================================
C     CALCULATE THE THETA BOUNDARY TEMPERATURES IN EACH THETA PLANE.
      DO 660 K=1,KM
      KG=KGT(K)
C     =================================================================
C     EXCLUDE THE NON-GAPLINES.
      IF (KG.EQ.0) GO TO 660
C     =================================================================
C     CALCULATE THE THETA BOUNDARY TEMPERATURES FOR EACH AXIAL COLUMN
C     IN THE PLANE.
```

```
      DO 650 I=2,IM
C     ===================================================================
C     CALCULATE THE THETA BOUNDARY TEMPERATURES IN THE COLUMN.
C     ===================================================================
      DO 640 J=2,JM
C     IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS
C    .POINT ? -'NO,YES'-
      IF (GAPT(KG,I,J).EQ.0.0) GO TO 640
C     IS IT A COOLANT OR A GAP WHICH IS ADJACENT ? -'COOLANT,GAP'-
      IF (GAPT(KG,I,J).LT.0.0) GO TO 510
C     ===================================================================
C     CALCULATE THE BOUNDARY TEMPERATURES WHEN A GAP IS ADJACENT.
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE TEM-TBBT ARRAYS.
C     BX IS (TEMH(KG,I,J)+TEML(KG,I,J)-1.0)
      BX = TBBTL(KG, I, J)
C     GK IS THE GAS THERMAL CONDUCTIVITY
      GK = TBBTH(KG, I, J)
C     CONL IS CONT(I,J,K)
      CONL = TEML(KG, I, J)
C     CONH IS CONT(I,J,K+1)
      CONH = TEMH(KG, I, J)
C     GET READY TO CALCULATE THE BOUNDARY TEMPERATURES OF THE GAP.
      B = TATIOB(KG, I, J)*CONL/CONH
      C = T(I, J, K) + B*T(I, J, K + 1)
      RGAP = 0.1713E-8*GAPT(KG, I, J)/BX
C     PUT RGAP IN THE CORRECT UNITS FOR CYLINDRICAL GEOMETRY.
      IF(ISHAPE.EQ.0) RGAP=RGAP*RP(I)
      ED=TATIOH(KG)*CONH/GAPT(KG,I,J)
C     ALLOW 10 ITERATIONS TO CALCULATE THE BOUNDARY TEMPERATURES.
      DO 500 KK=1,10
      IF (KK.EQ.1) GO TO 480
      TH = TBBTH(KG, I, J)
      TB = TBBTL(KG, I, J)
      GO TO 490
  480 TH = T(I, J, K)
      TB = T(I, J, K + 1)
  490 GH = (GK + RGAP*(TH**2 + TB**2)*(TH + TB))*ED
      X1 = GH*C
      X2 = 1.0 + GH*(B + 1.0)
      TBBTL(KG, I, J) = (X1 + T(I, J, K + 1))/X2
      TBBTH(KG, I, J) = (X1 + T(I, J, K))/X2
C     ARE THE BOUNDARY TEMPERATURES CONVERGED ? -'YES,NO'-
      IF (ABS(TBBTH(KG,I,J)-TH).LE.1.0) GO TO 640
C     HAVE THE 10 ITERATIONS BEEN PERFORMED ? -'YES,NO'-
  500 CONTINUE
```

```
C     NEGATIVE TEMPERATURES ARE PRESENT AND RESULT IN LOCAL
C    .INSTABILITIES. TRY A SMALLER TIME STEP.
      IERROR(1) = ISURT(1)
      IERROR(2)=ISURT(4)
      CALL ERROR2
C     =================================================================
C     CALCULATE THE BOUNDARY TEMPERATURES WHEN A COOLANT IS ADJACENT.
C     IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE ? -,NO,YES'-
  510 IF (GAPT(KG,I,J).LT.(-1.5E-10)) GO TO 570
C     CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE HIGH
C    .INDEX SIDE.
      DXT=TL(K)-TP(K)
C     PUT DXT IN THE CORRECT UNITS FOR CYLINDRICAL GEOMETRY.
      IF(ISHAPE.EQ.0) DXT=DXT*RP(I)
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEDING ITERATION AND PRESERVED IN THE TEM-TBBT ARRAYS.
C     CONL IS CONT(I,J,K)
      CONL = TEML(KG, I, J)
C     CONH IS CONT(I,J,K+1)
      CONH = TBBTL(KG, I, J)
C     EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.
      IF (KG.EQ.NTG.OR.MATTG(KG,I,J).NE.200) GO TO 550
C     FIND THE HIGH THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      KG1=KG+1
      DO 530 KK=KG1,NTG
      IF (GAPT(KK,I,J).LT.(-1.5E-10)) GO TO 520
      GO TO 530
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C     THE HIGH THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  520 IF (GAPT(KK,I,J).LT.(-2.5E-10)) GO TO 550
      KGH=KK
      GO TO 540
  530 CONTINUE
  540 RAD = 0.1713E-8*(TBBTH(KG, I, J)**4 - TBBTL(KGH, I, J)**4)/(TEMH(
     ' KG, I, J) + TEML(KGH, I, J) - 1.0)
C     PRESERVE RAD IN THE TEML ARRAY.
      TEML(KG, I, J) = RAD
      HX = CONH*(T(I, J, K) - T(I, J, K + 1) - RAD*CONL*DXT)/(T(I, J, K)
     '  - T(I, J, K + 1) + RAD*CONH)
      GO TO 560
  550 HX = CONH
C     CALCULATE NEW BOUNDARY TEMPERATURES.
  560 TBBTH(KG, I, J) = (CONL*T(I, J, K + 1) + HX/DXT*T(I, J, K))/(CONL
     '  + HX/DXT)
      TBBTL(KG, I, J) = TBBTH(KG, I, J)
    ' GO TO 640
```

```
C     ARE TWO COOLANTS ADJACENT ? -'NO,YES'-
  570 IF (GAPT(KG,I,J).GT.(-2.5E-10)) GO TO 580
C     DO NOT ATTEMPT TO CALCULATE BOUNDARY TEMPERATURES.
      GO TO 640
C     CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE LOW
C    .INDEX SIDE.
  580 DXT=TP(K+1)-TL(K)
C     PUT DXT IN THE CORRECT UNITS FOR CYLINDRICAL GEOMETRY.
      IF(ISHAPE.EQ.0) DXT=DXT*RP(I)
C     RECOVER PARAMETERS WHICH WERE EVALUATED FROM TEMPERATURES OF THE
C     PRECEEDING ITERATION AND PRESERVED IN THE TEM-TBBT ARRAYS.
C     CONL IS CONT(I,J,K)
      CONL=TBBTH(KG,I,J)
C     CONH IS CONT(I,J,K+1)
      CONH=TEMH(KG,I,J)
C     EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.
      IF (KG.EQ.1) GO TO 620
      IF (NATTG(KG,I,J).NE.200) GO TO 620
C     FIND THE LOW THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.
      KG1=KG-1
      DO 600 KK=1,KG1
      KG2=KG-KK
      IF (GAPT(KG2,I,J).LT.(-.5E-10)) GO TO 590
      GO TO 600
C     EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT
C     THE LOW THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.
  590 IF (GAPT(KG2,I,J).LT.(-2.5E-10)) GO TO 620
      KGL=KG2
      GO TO 610
  600 CONTINUE
C     RECOVER THE VALUE OF RAD PRESERVED IN THE TEML ARRAY.
  610 RAD = TEML(KGL, I, J)
      HX = CONL*(T(I, J, K + 1) - T(I, J, K) + CONH*RAD*DXT)/(T(I, J, K
     * + 1) - T(I, J, K) - CONL*RAD)
      GO TO 630
  620 HX=CONL
C     CALCULATE NEW BOUNDARY TEMPERATURES.
  630 TBBTL(KG, I, J) = (CONH*T(I, J, K) + HX/DXT*T(I, J, K + 1))/(CONH
     * + HX/DXT)
      TBBTH(KG, I, J) = TBBTL(KG, I, J)
C     ===================================================================
C     HAVE ALL POINTS ALONG THE CURRENT THETA GAPLINE BEEN
C    .CONSIDERED ? -'YES,NO'-
  640 CONTINUE
C     ===================================================================
C     HAVE ALL COLUMNS IN THE PLANE BEEN CONSIDERED ? -'YES,NO'-
  650 CONTINUE
C     ===================================================================
C     HAVE ALL THETA BOUNDARY TEMPERATURES BEEN CALCULATED ? -'YES,NO'-
  660 CONTINUE
C     ===================================================================
      RETURN
      END
```

```
      SUBROUTINE PRINT
      INCLUDE       COMDIM
C=======================================================================
C=======================================================================
      INTEGER           TT
      LOGICAL          WRES
      DIMENSION         DTO    ( NQ),DTI    ( NQ),WAY    (  6),
     1                  AT     ( IQ,JQ,KQ),          CT     ( IQ,JQ,KQ),
     3                  DDT    ( IQ,JQ,KQ)
      DATA (WAY(I),I=1,6)  /6H+1,J,K, 6H,J+1,K, 6H,J,K+1, 3H(I), 3H(J),
     -                     3H(K)/
      EQUIVALENCE       (TT     ,AT     ),
     1                  (TT     ,CT     ),(TT     ,DDT    ),(ITO    ,DTO    ),
     2                  (ITI    ,DTI    )
C=    ==================================================================
C=    ==================================================================
      IRET=0
      WRES=.FALSE.
      X1=CURTI*60.0
      X2=CURTI*3600.0
C     SELECT THE TEMPERATURE DISTRIBUTION PRINT FORMAT
      IF(LASTIP.EQ.NITER) GO TO 190
      IF (SW(9)) GO TO 190
C=    ==================================================================
C     PREPARE AND PRINT COOLANT AND NODE TEMPERATURE DISTRIBUTION IN
C     INTEGRAL DEGREES FAHRENHEIT
      GO TO 167
      ENTRY TEMPS
      IRET=1
  167 WRITE(6,110) (ZA(I),I=1,12)
  110 FORMAT (1H1,30X,12A6,///)
      IF(IRET.EQ.2) GO TO 169
      WRITE (6,111)
  111 FORMAT                      (49X,28HCOOLANT  TEMPERATURES    (F),/,3X,
     114HCOOLANT NUMBER,4X,5HINLET,5X,6HOUTLET,4X,12HFLOW (LB/HR),8X,14H
     2COOLANT NUMBER,4X,5HINLET,5X,6HOUTLET,4X,12HFLOW (LB/HR))
C     PREPARE ALL COOLANT DATA FOR PRINTING
      DO 130 N=1,MAXFLO,2
      IF (NFLO(N).EQ.0) GO TO 150
      N1=NFLO(N)
      N2=NFLO(N+1)
      ITI(N1)=TI(N1)-459.5
      ITO(N1)=TO(N1)-459.5
      IF(N1)=FLOW(N1)
      IF (N2.EQ.0 .OR. N.EQ.15) GO TO 120
```

```
      ITI(N2)=TI(N2)-459.5
      ITO(N2)=TO(N2)-459.5
      IF(N2)=FLOW(N2)
      WRITE (6,140) N1,ITI(N1),ITO(N1),IF(N1),N2,ITI(N2),ITO(N2),IF(N2)
  140 FORMAT(8X,I2,I16,I11,I14,15X,I2,I16,I11,I14)
      GO TO 130
  120 WRITE (6,140) N1,ITI(N1),ITO(N1),IF(N1)
      GO TO 150
  130 CONTINUE
  150 IF(IRET.EQ.0)  GO TO 153
  169 WRITE(6,155) NITER
  155 FORMAT(1H0,3AX,I4,51H ITERATIONS PERFORMED WITHOUT REACHING STEADY
     . STATE,/)
      IF(IRET.EQ.2) RETURN
      GO TO 157
  153 WRITE(6,160) CURTI,X1,X2,NITER
  160 FORMAT (1H0,21H  THE CURRENT TIME IS,F10.4,11H HOURS   =   ,F10.4,13
     1H MINUTES   =   ,F13.5,8H SECONDS,4X,I4,31H ITERATIONS HAVE BEEN PER
     2FORMED,/)
C=    ===================================================================
C     CONVERT TEMPERATURES TO INTEGRAL DEGREES FAHRENHEIT
  157 DO 170 I=1,IMAX
      DO 170 J=1,JMAX
      DO 170 K=1,KMAX
      TT(I,J,K)=T(I,J,K)-459.5
  170 CONTINUE
C     SET THE MESH RIBS IN THE PRINTOUT EQUAL TO ZERO
      DO 516 L = 1,JMAX
      TT(1,L,1) = 0
      TT(IMAX,L,1) = 0
      TT(1,L,KMAX) = 0
  516 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 553 L = 1,KMAX
      TT(1,1,L) = 0
      TT(IMAX,1,L) = 0
      TT(1,JMAX,L) = 0
  553 TT(IMAX,JMAX,L) = 0
C=========================================================================
C=    ===================================================================
C     PRINT THE TEMPERATURE (INTEGRAL DEGREES F)
      WRITE (6,180)
```

```
  160 FORMAT (1H0,47X,20HTEMPERATURES      (F))                     PRIN 920
      CALL ARRAY (TT,DUM,1)                                         PRIN 930
      IF(IRET.EQ.0) GO TO 159                                       PRIN 940
      IRET=2                                                        PRIN 950
      GO TO 167                                                     PRIN 960
  159 IF(IERROR(1).GT.0) WRITE(6,185) IERROR(1),NITER               PRIN 970
  185 FORMAT(1H0,5H*****,I4,52H ERROR MESSAGES PRECEED THE RESULTS OF ITPRIN 980
     .ERATION NO.,I4,5H*****)                                       PRIN 990
      LASTIP=NITER                                                  PRIN1000
      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
     1N2)                                                           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,IMAX                                               PRIN1340
      DO 260 J=1,JMAX                                               PRIN1350
      DO 260 K=1,KMAX                                               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
  260 CONTINUE
C     SET THE MESH RIBS IN THE PRINTOUT EQUAL TO ZERO
      DO 501 L = 1,JMAX
      DDT(1,L,1)=0.0
      DDT(IMAX,L,1)=0.0
      DDT(1,L,KMAX)=0.0
  501 DDT(IMAX,L,KMAX)=0.0
      DO 526 L=1,IMAX
      DDT(L,1,1)=0.0
      DDT(L,JMAX,1)=0.0
      DDT(L,1,KMAX)=0.0
  526 DDT(L,JMAX,KMAX)=0.0
      DO 551 L=1,KMAX
      DDT(1,1,L)=0.0
      DDT(IMAX,1,L)=0.0
      DDT(1,JMAX,L)=0.0
  551 DDT(IMAX,JMAX,L)=0.0
C==========================================================================
C=    ====================================================================
C     PRINT THE TEMPERATURES (DECIMAL DEGREES F)
      WRITE (6,270)
  270 FORMAT (1H0,47X,28HDECIMAL TEMPERATURES      (F))
      CALL ARRAY (IDUM,DDT,3)
      IF(IERROR(1).GT.0) WRITE(6,185)IERROR(1),NITER
C     ====================================================================
C     CALCULATE AND PRINT THE RESIDUALS FOR THE LAST ITERATION OF A
C     STEADY STATE PROBLEM.
      IF(LP.AND.SW(10)) GO TO 273
      GO TO 280
  273 WRES=.TRUE.
      GO TO 355
  275 CALL RESID
      WRES=.FALSE.
C=    ====================================================================
C     DO WHAT THE USER WISHES TO DO
  280 CALL CUSTOM
C=    ====================================================================
C     SHOULD THE SURFACE TEMPERATURES BE PRINTED? -,NO,YES,-
      IF (.NOT.SW(8)) GO TO 340
C=    ====================================================================
C     PRINT THE RADIAL-X SURFACE TEMPERATURES (LOW AND HIGH)
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,290) PNAME(1),PNAME(2),WAY(6),PNAME(3),WAY(5)
  290 FORMAT              (19X,32HSURFACE  TEMPERATURES  AT  THE   A6,
     '34BH  GAP  OR  COOLANT  BOUNDARY  GRIDPLANES      (F),//,43X,4HTHE ,
```

```
     XA6,A3,32H DIRECTION POINTS ARE HORIZONTAL/43X,4HTHE ,A6,A3,30H DIRPRIN1840
     XECTION 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 ANDHIGH)              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,GAPT)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(WRES) GO TO 275                                                PRIN2240
      WRITE (6,370) PNAME(1),WAY(1)                                     PRIN2250
  370 FORMAT (     19X,15HHEAT RATE  IN ,A6,46H  DIRECTION  BETWEEN  POIPRIN2260
     1NTS  (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
      NAML=ISHAPE+1
      DO 410 K=1,KMAX
      DO 410 J=1,JMAX
      DO 410 I=1,IM
      GO TO (380,390),NAME
  380 AT(I,J,K)=AT(I,J,K)/(RR(I,J,K)*RL(I)*ALOG(RP(I+1)/RP(I)))
      GO TO 410
  390 AT(I,J,K)=AT(I,J,K)/(RR(I,J,K)*DELR(I))
  410 CONTINUE
C     PRINT THE  RADIAL-X  HEAT FLUXES
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,420) PNAME(1),WAY(4),WAY(1)
  420 FORMAT (     19X,16HHEAT  FLUX  IN   ,A6,48H  DIRECTION  BASED  ON
     1THE  AREA  OF  GRIDLINE ,A3,/,28X,33HBETWEEN  POINTS  (I,J,K)  AND
     2  (I,A6,20H)       (BTU/HR-FT**3))
      CALL ARRAY (IDUM,AT,2)
C=    ==================================================================
C     CALCULATE THE  AXIAL-Z  HEAT RATE
      DO 430 K=1,KMAX
      DO 430 J=1,JMAX
      DO 430 I=1,IMAX
      AT(I,J,K)=KZ(I,J,K)*KZ(I,J,K)*(T(I,J,K)-T(I,J+1,K))
  430 CONTINUE
C     PRINT THE  AXIAL-Z  HEAT RATE
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,370) PNAME(2),WAY(2)
      CALL ARRAY (IDUM,AT,2)
C=    ==================================================================
C     CALCULATE THE  AXIAL-Z  HEAT FLUXES
      DO 480 J=1,JM
      DO 480 I=1,IMAX
      DO 480 K=1,KMAX
      GO TO (450,460),NAME
  450 AT(I,J,K)=AT(I,J,K)/(RZ(I,J,K)*(ZP(J+1)-ZP(J)))
      GO TO 480
  460 AT(I,J,K)=AT(I,J,K)/(RZ(I,J,K)*DELZ(J))
  480 CONTINUE
C     PRINT THE  AXIAL-Z  HEAT FLUXES
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,420) PNAME(2),WAY(5),WAY(2)
      CALL ARRAY (IDUM,AT,2)
C=    ==================================================================
C=    ==================================================================
C     CALCULATE THE  THETA-Y  HEAT RATE
      DO 1430 K=1,KMAX
```

```
      DO 1430 J=1,JMAX
      DO 1430 I=1,IMAX
      AT(I,J,K)=RT(I,J,K)*KT(I,J,K)*(T(I,J,K)-T(I,J,K+1))
 1430 CONTINUE
C     PRINT THE  THETA-Y  HEAT RATE
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,370) PNAME(3),WAY(3)
      CALL ARRAY (IDUM,AT,2)
C=    ==================================================================
C     CALCULATE THE THETA-Y  HEAT FLUXES
      DO 1480 K=1,KM
      DO 1480 J=1,JMAX
      DO 1480 I=1,IMAX
      GO TO (1450,1460),NAME
 1450 AT(I,J,K)=AT(I,J,K)/(RT(I,J,K)*(RP(I)*(TP(K+1)-TP(K))))
      GO TO 1480
 1460 AT(I,J,K)=AT(I,J,K)/(RT(I,J,K)*DELT(K))
 1480 CONTINUE
C     PRINT THE  THEATA-Y  HEAT FLUXES
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,420) PNAME(3), WAY(6),WAY(3)
      CALL ARRAY (IDUM,AT,2)
C=    ==================================================================
C     SHOULD THE EFFECTIVE THERMAL CONDUCTIVITIES AND THERMAL
C    -CONDUCTANCES BE PRINTED? -'NO,YES'-
  500 IF (.NOT.SW(7)) GO TO 610
C=    ==================================================================
      DO 510 K=1,KMAX
      DO 510 I=1,IMAX
      KR(I,1,K)=0.0
  510 KR(I,JMAX,K)=0.0
      DO 515 I=1,IMAX
      DO 515 J=1,JMAX
      KR(I,J,1)=0.0
  515 KR(I,J,KMAX)=0.0
      DO 520 K=1,KMAX
      DO 520 J=1,JMAX
  520 KR(IMAX,J,K)=0.0
C     PRINT THE EFFECTIVE  RADIAL-X  THERMAL CONDUCTIVITIES
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,530) PNAME(1),WAY(1)
  530 FORMAT (     19X,11HEFFECTIVE  ,A6,49H CONDUCTIVITY  BETWEEN  POIN
     1TS  (I,J,K)  AND  (I,A6,19H)     (BTU/HR-FT-F))
      CALL ARRAY (IDUM,KR,2)
C=    ==================================================================
      DO 540 K=1,KMAX
```

```
      DO 540 I=1,IMAX
  540 KZ(I,JMAX,K)=0.0
      DO 545 I=1,IMAX
      DO 545 J=1,JMAX
      KZ(I,J,1)=0.0
  545 KZ(I,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 (6,530) PNAME(2),WAY(2)
      CALL ARRAY (IDUM,KZ,2)
C=    =================================================================
      DO 1540 I=1,IMAX
      DO 1540 J=1,JMAX
 1540 KT(I,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(I,1,K)=0.0
 1550 KT(I,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 (6,530) 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(I,J,K)=RR(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 (6,580) 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
```

```
      CT(I,J,K)=RZ(I,J,K)*KZ(I,J,K)                                     PRIN3680
  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)                                     PRIN3780
 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.SW(3)) GO TO 620                                         PRIN3860
C=    ===================================================================PRIN3870
C     PUT THE CURRENT TEMPERATURE DISTRIBUTION ON TAPE                  PRIN3880
      WRITE (OUTTAP) 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. DP) GO TO 640                                           PRIN3940
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER                    PRIN3950
      WRITE (6,1000)                                                    PRIN3960
 1000 FORMAT (52X,13HHEAT  BALANCE,///,29X,15HHEAT  GENERATED,25X,      PRIN3970
     X28HHEAT LOST/GAINED BY COOLANTS, //, 25X,12HBLOCK NUMBER,9X,      PRIN3980
     X6HBTU/HR,13X,14HCOOLANT NUMBER,8X,6HBTU/HR,//)                    PRIN3990
      SUMM=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
      I1=IABS(IPATH(N))                                                 PRIN4130
```

```
      GO TO (1025,1050,1010), II
C     AXIAL-Z COOLANT FLOW
 1050 DO 1060 J=JLS,JHS
      DO 1065 K=KLS,KHS
      IF (ILS .LE. 2) GO TO 1055
      I=ILS-1
      SUM =SUM + RR(I,J,K) * KR(I,J,K) *(T(I,J,K)-T(ILS,J,K))
 1055 IF (IHS .GE. IM) GO TO 1065
      SUM = SUM - RR(IHS,J,K) * KR(IHS,J,K) *(T(IHS,J,K) - T(IHS+1,J,K))
 1065 CONTINUE
      DO 1068 I=ILS,IHS
      IF (KLS .LE. 2) GO TO 1069
      K=KLS-1
      SUM=SUM + RT(I,J,K) * KT(I,J,K) * (T(I,J,K) - T(I,J,KLS))
 1069 IF (KHS .GE.KM) GO TO 1068
      SUM = SUM - RT(I,J,KHS) * KT(I,J,KHS) *(T(I,J,KHS) - T(I,J,KHS+1))
 1068 CONTINUE
 1060 CONTINUE
      GO TO 1075
C     RADIAL-X COOLANT FLOW
 1025 DO 1040 I=ILS,IHS
      DO 1030 K=KLS,KHS
      IF (JLS .LE. 2) GO TO 1028
      J=JLS-1
      SUM = SUM + RZ(I,J,K) * KZ(I,J,K) * (T(I,J,K) - T(I,JLS,K))
 1028 IF (JHS .GE. JM) GO TO 1030
      SUM = SUM -RZ(I,JHS,K) * KZ(I,JHS,K) * (T(I,JHS,K)-T(I,JHS+1,K))
 1030 CONTINUE
      DO 1035 J=JLS,JHS
      IF (KLS .LE. 2) GO TO 1032
      K=KLS-1
      SUM = SUM + RT(I,J,K) * KT(I,J,K) * (T(I,J,K) - T(I,J,KLS))
 1032 IF (KHS .GE. KM) GO TO 1035
      SUM = SUM - RT(I,J,KHS)*KT(I,J,KHS)*(T(I,J,KHS)-T(K,J,KHS+1))
 1035 CONTINUE
 1040 CONTINUE
      GO TO 1075
C     THETA-Y COOLANT FLOW
 1010 DO 1020 K=KLS,KHS
      DO 1015 J=JLS,JHS
      IF (ILS .LE. 2) GO TO 1012
      I=ILS-1
      SUM=SUM + RR(I,J,K) * KR(I,J,K) * (T(I,J,K)-T(ILS,J,K))
 1012 IF (IHS.GE. IM) GO TO 1015
      SUM = SUM-RR(IHS,J,K) * KR(IHS,J,K) * (T(IHS,J,K)-T(IHS+1,J,K))
 1015 CONTINUE
```

```
      DO 1024 I=ILS,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 1160 I=ILS,IHS
      DO 1160 J=JLS,JHS
      DO 1160 K=KLS,KHS
      SUM = SUM + W(I,J,K)
 1160 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     ==================================================================
  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
C=    ==================================================================
C     ******************************************************************
C     ******************************************************************
      SUBROUTINE GPRINT(NGAP,NGRID,PLABL,ICROSS,IDOWN,N1,N2,N3,BBTH,
     XBBTL,GAP)
C=    ==================================================================
      DIMENSION        NGAP  (  1), BBTH(N1,N2,N3), BBTL(N1,N2,N3),
     X                 GAP(N1,N2,N3)
      DIMENSION        FORMR ( 12),RPRINT(  4),FORMT ( 12),ITT   (20)
      DATA FORMR      /4H(I3,, 4H12X,, 9*4H13X,, 1H)/
      DATA RPRINT     /6HI6,I6,, 6HI7,I6,, 6HI8,4X,, 6HI9,4X,/
      DIMENSION        NUMBER( 25)
      DATA NUMBER     /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
     1                 20,21,22,23,24,25/
```

```
      I=0
      DO 311 M=1,NGRID
      IF (NGAP(M).EQ.0) GO TO 311
      I=I+1
      WRITE (6,312) PLABL,M
312   FORMAT (1H0,53X,A6,2H (,I2,11H) GRIDPLANE,/)
      NUM2=0
292   NUM1=NUM2+1
      NUM2=NUM2+10
      IF (NUM2.GT.ICROSS) NUM2=ICROSS
      WRITE (6,293) (NUMBER(L), L=NUM1,NUM2)
293   FORMAT (1H0,110,9I13)
      WRITE (6,294)
294   FORMAT (1H0)
      DO 310 K=1,IDOWN
      N=0
      NCOUNT=1
      DO 301 MM=1,12
301   FORMT(MM)=FORMR(MM)
      DO 300 J=NUM1,NUM2
      NCOUNT=NCOUNT+1
      IT1= BBTH(I,J,K)-459.5
      IF (IT1.LE.-450) GO TO 300
      IT2= BBTL(I,J,K)-459.5
      IF (IT1.EQ.IT2 .AND. GAP (I,J,K).LT.0.0) GO TO 302
      N=N+2
      ITT(N-1)=IT1
      ITT(N)=IT2
      FORMT(NCOUNT)=RPRINT(2)
      IF (NCOUNT.EQ.2) FORMT(NCOUNT)=RPRINT(1)
      GO TO 300
302   N=N+1
      ITT(N)=IT1
      FORMT(NCOUNT)=RPRINT(4)
      IF (NCOUNT.EQ.2) FORMT(NCOUNT)=RPRINT(3)
300   CONTINUE
      IF (N.EQ.0) GO TO 309
      WRITE (6,FORMT) K,(ITT(L),L=1,N)
      GO TO 310
309   WRITE (6,FORMT) K
310   CONTINUE
      IF (NUM2.LT.ICROSS) GO TO 292
311   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ARRAY (IX,X,IFY)
      INCLUDE      COMDIM
C==========================================================================
CB    PRINT A THREE-DIMENSIONAL ARRAY
C==========================================================================
C==========================================================================
      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/
C==========================================================================
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
      N=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
      N=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
      N=IMAX
   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)
```

```
  221 WRITE (6,170)
  170 FORMAT (1H0)
      GO TO (200,300,400),NPRINT
C======================================================================
C     OUTPUT PRINT IN RADIAL-X DIRECTION, THETA-Y DIRECTION HORIZONTAL,
C     AXIAL-Z DIRECTION VERTICAL
C======================================================================
  200 DO 420 J=1,JMAX
      GO TO (421,422,423),IFY
  421 WRITE (6,203) J,(IX(M,J,K),K=NUM1,NUM2)
  203 FORMAT (I4,1X,I6,24I5)
      GO TO 420
  422 WRITE (6,201) J,(X(M,J,K),K=NUM1,NUM2)
  201 FORMAT (I4,1P10E12.4)
      GO TO 420
  423 WRITE (6,202) J,(X(M,J,K),K=NUM1,NUM2)
  202 FORMAT (I4,1P10E12.5)
  420 CONTINUE
      IF(NUM2 .LT. N) GO TO 100
      GO TO 600
C======================================================================
C     OUTPUT PRINT IN AXIAL-Z DIRECTION, RADIAL-X DIRECTION HORIZONTAL,
C     THETA-Y DIRECTION VERTICAL
C======================================================================
  300 DO 320 K=1,KMAX
      GO TO (321,322,323),IFY
  321 WRITE (6,203) K,(IX(I,M,K), I=NUM1,NUM2)
      GO TO 320
  322 WRITE (6,201) K,(X(I,M,K), I=NUM1,NUM2)
      GO TO 320
  323 WRITE (6,202) K,(X(I,M,K), I=NUM1,NUM2)
  320 CONTINUE
      IF(NUM2 .LT. N) GO TO 100
      GO TO 600
C======================================================================
C     OUTPUT PRINT IN THETA-Y DIRECTION, RADIAL-X DIRECTION HORIZONTAL,
C     AXIAL-Z DIRECTION VERTICAL
C======================================================================
  400 DO 230 J=1,JMAX
      GO TO (231,232,233), IFY
  231 WRITE (6,203) J,(IX(I,J,M),I=NUM1,NUM2)
      GO TO 230
  232 WRITE (6,201) J,( X(I,J,M),I=NUM1,NUM2)
      GO TO 230
  233 WRITE (6,202) J,( X(I,J,M),I=NUM1,NUM2)
  230 CONTINUE
      IF(NUM2 .LT. N) GO TO 100
  600 CONTINUE
      RETURN
C======================================================================
      END
```

```
      SUBROUTINE CUSTOM
      INCLUDE     COMDIM
C=======================================================================
C     DO WHAT THE USER WANTS TO DO
C=======================================================================
C=======================================================================
C=======================================================================
C=======================================================================
C=======================================================================
C=======================================================================
      RETURN
      END
```

```
      SUBROUTINE PUN
      INCLUDE       COMDIM
C     ==================================================================
C     PUNCH THE FINAL TEMPERATURE DISTRIBUTION.
C     ==================================================================
      DIMENSION      ALABEL(4), TPUN(1)
      EQUIVALENCE     (TT    ,TPUN  )
      DATA ALABEL     /6HHEADER, 5HTEMP , 5HINLET, 5HOUTLT/
C     ==================================================================
C     ==================================================================
      WRITE (6,210) (ZA(I),I=1,12)
      IF(.NOT.SW(10)) GO TO 300
      CURTI=0.0
      NITER=0
  300 BIMAX=IMAX
      BJMAX=JMAX
      BKMAX=KMAX
      BNITER=NITER
      WRITE (6,220) CURTI,BIMAX,BJMAX,BKMAX,BNITER,ALABEL(1)
      PUNCH 230, CURTI,BIMAX,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
     1DECK))
C=    ==================================================================
CE    ******************************************************************
CE    ******************************************************************
CE    ******************************************************************
```

```
      SUBROUTINE PUNCHY (TP,LENGTH,ANAME)
C=    ==================================================================
C     PUNCH THE TEMPERATURE DISTRIBUTION DECK
C=    ==================================================================
C=    ==================================================================
      DIMENSION        TP     (  1),FORM   (  9),FORMT (  9),FB     (  3)
      DATA FORMT       /6H       , 6H     1P, 6*6HE12.6,, 6HA5,I3)/
      DATA FB          /6H( 10X,, 6H(       , 6H  12X,/
C=    ==================================================================
      N=LENGTH/6
      NLEFT=LENGTH-N*6
      NN1=1
      NN2=6
      DO 90 I=1,9
      FORM(1)=FORMT(I)
   90 CONTINUE
      DO 100 KK=1,N
      FORM(1)=FB(1)
      WRITE (6,FORM) (TP(I),I=NN1,NN2),ANAME,KK
      FORM(1)=FB(2)
      PUNCH FORM, (TP(I),I=NN1,NN2),ANAME,KK
      NN1=NN2+1
      NN2=NN2+6
  100 CONTINUE
C=    ==================================================================
      IF (NLEFT.EQ.0) GO TO 120
      KK=N+1
      J=NLEFT+3
      DO 110 I=J,8
      FORM(1)=FB(3)
  110 CONTINUE
      FORM(1)=FB(1)
      WRITE (6,FORM) (TP(I),I=NN1,LENGTH),ANAME,KK
      FORM(1)=FB(2)
      PUNCH FORM, (TP(I),I=NN1,LENGTH),ANAME,KK
C=    ==================================================================
  120 RETURN
      END
```

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.