29
2-24-70

DRAGON Project

MASTER

Gulf General Atomic
Incorporated

AEC RESEARCH AND
DEVELOPMENT REPORT

GA-9262
UC-32, Mathematics
and Computers

TAC2D
A GENERAL PURPOSE TWO-DIMENSIONAL
HEAT TRANSFER COMPUTER CODE
MATHEMATICAL FORMULATIONS AND
PROGRAMMER'S GUIDE

by
S. S. Clark and J. F. Petersen

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

September 1969

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

LEGAL NOTICE

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

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

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

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

Printed in the United States of America

Available from

Clearinghouse for Federal Scientific and Technical Information

National Bureau of Standards, U. S. Department of Commerce

Springfield, Virginia 22151

Price: Printed Copy \$3.00; Microfiche \$0.65

DISCLAIMER

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

DISCLAIMER

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

Gulf General Atomic Incorporated

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

AEC RESEARCH AND
DEVELOPMENT REPORT

GA-9262

TAC2D A GENERAL PURPOSE TWO-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. F. Petersen

NOTICE

This report was received under the provisions
of the *San Diego Project*
arrangement and is subject to the terms thereof.

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

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission
A. Makes any warranty or representation, expressed or implied with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights, or
B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report
As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Gulf General Atomic Project 317

September 1969

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

The TAC2D computer code is described herein as it existed on September 1969. The code has been in continuous development for 2 years and in its presented form has been applied successfully by Gulf General Atomic Incorporated to the kind of problems discussed later in this report. However, the development and improvement of the code are being continued, so that duplication of results (or even close agreement) between problems run with the code as published and the code as it existed either before or after this time is not necessarily to be expected.

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

ABSTRACT

TAC2D is a code for calculating steady-state and transient temperatures in two-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, cylindrical, or circular (polar) coordinate system by orthogonal lines of constant coordinate called grid lines. The grid lines specify an array of nodal elements. Nodal points are defined as lying midway between the bounding grid lines 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 line system must be orthogonal in the rectangular, cylindrical, or circular coordinate system. Therefore, the sides of the nodal elements must also be orthogonal. The entire problem must be bounded by four grid lines 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.

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

TABLE OF CONTENTS

ABSTRACT	iii
TABLE OF CONTENTS	v
LIST OF FIGURES	vi
LIST OF TABLES	vii
1. INTRODUCTION	1
2. GEOMETRICAL CONSIDERATIONS	5
2.1 Grid Lines, Points, and Boundaries	5
2.2 Blocks	9
2.3 Gaps	12
2.4 Gap Lines	13
3. MATERIAL, GAS AND COOLANT THERMAL PARAMETERS	15
3.1 Description of Thermal Parameter Functions	15
3.2 Evaluation of Thermal Parameters	16
3.3 Function Control Constants	17
4. GENERAL CODE DESCRIPTION	19
4.1 Program Sections	19
APPENDIX A - MATHEMATICAL SOLUTION	23
APPENDIX B - GEOMETRY FACTORS	27
APPENDIX C - COMPOSITE CONDUCTIVITIES	31
C.1 No Gap Between Adjacent Mesh Points, Equal Thermal Conductivities	31
C.2 No Gap Between Adjacent Mesh Points, Unequal Thermal Conductivities	32
C.3 A Gap Between Adjacent Mesh Points	35
C.4 Coolants	40
APPENDIX D - COOLANTS	47
APPENDIX E - STEADY-STATE OPTION	53

Table of Contents (Continued)

APPENDIX F - SAMPLE PROBLEM	57
F.1 Problem Description	57
F.2 Analytical Solution	58
F.3 Numerical Solution with TAC2D	58
F.4 Comparison of the TAC2D Solution with the Analytical Solution	60
APPENDIX G - CODE ORGANIZATION AND FUNCTIONS OF SUBROUTINES IN TAC2D .	67
APPENDIX H - TAC2D CODE LISTING	75
REFERENCES	215

LIST OF FIGURES

Fig. 1 Cylindrical geometry envelope showing grid lines and mesh points	6
Fig. 2 Circular geometry envelope showing grid lines and mesh points	7
Fig. 3 Rectangular geometry envelope showing grid lines and mesh points	8
Fig. 4 Typical problem model in circular geometry	10
Fig. 5 General flow of logic in TAC2D	21
Fig. C.1 Logical search at each grid plane	32
Fig. F.1 TAC2D sample problem, thermal model	59
Fig. F.2 Transient temperatures in a right circular cylinder - comparison of TAC2D results with series solution	65

LIST OF TABLES

Table 1	Block Boundary and Block Point Nomenclature	9
Table 2	Internal Block Definition	11
Table 3	External Block Definition	11
Table C.1	Geometrical Constant $(\Delta X_1/A_1)/(\Delta X/A)$	34
Table C.2	Geometrical Constant $(A_2/\Delta X_2)/(A/\Delta X)$	38
Table C.3	Geometrical Constant $(A_2/\Delta X_2)/(A_1/\Delta X_1)$	39
Table C.4	Geometrical Constant $A_g/(A_2/\Delta X_2)$	39
Table C.5	Geometrical Constant $(A_1^*/\Delta X_1^*)/A_1$	44
Table C.6	Geometrical Constant $(A_2^*/\Delta X_2^*)/A_2$	45
Table C.7	Geometrical Constant $(A_1^{**}/X)/A_1$ or $(A_2^{**}/X)/A_2$	45
Table C.8	Geometrical Constant A_1/A_2	46
Table F.1	Input Thermal Parameter Functions for TAC2D Sample Problem	60
Table F.2	Excerpts from TAC2D Sample Problem Output Showing Block, Grid Plane and Time History Specifications	61

1. INTRODUCTION

The purpose of this report is to document methods of solution and programming logic for TAC2D*. All information required for use of the code is given in Ref. 1.

The TAC2D code is designed to treat transient, two-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), cylindrical (R, Z), or circular (R, Θ) coordinates. Although there are a number of two-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 gradient scheme which requires a "regular" geometry in that the points at which temperatures are to be calculated must be in regular rows and columns. As a consequence, TAC2D is primarily suited to solve problems that roughly fit an envelope of either a rectangle, a right circular cylinder, or a polar rectangle.

*The acronym TAC2D stands for "Thermal Analysis Code, Two-Dimensional."

The basic heat transport equation to be solved is

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

where

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

T is local temperature, °F

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

ρ is density, lb/ft³

c is specific heat, Btu/lb-°F

τ is time, hr

This equation is replaced by its equivalent set of linear finite difference equations. This set of equations is then solved by an implicit alternating gradient 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 TAC2D. 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 TAC2D code has great versatility because of its ability to accept material, gas and coolant thermal parameters in a functional form. Axial conductivity, radial 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 D of Ref. 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.

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

2. GEOMETRICAL CONSIDERATIONS

2.1 GRID LINES, POINTS, AND BOUNDARIES

Three geometry types are available to describe the model — cylindrical, circular, and rectangular. The model envelope and coordinate axes are determined from this choice.

<u>Geometry Type</u>	<u>Model Envelope</u>	<u>Coordinate Axes</u>
Cylindrical	Hollow or solid right circular cylinder	R, Z
Circular	Polar rectangle* ($0 \leq \theta \leq 2\pi$)	R, θ
Rectangular	Rectangle	X, Y

From this point on the words "radial" and "axial" will be used for general reference to the coordinate axes. Unless specifically noted, usage of the word "axial" will imply reference to the theta coordinate of the circular geometry type as well as to the axial coordinate of the cylindrical geometry type. For the rectangular geometry type, "X" and "Y" may be equated with "radial" and "axial" in either order.

In each of the coordinate directions, a set of grid lines, not necessarily evenly spaced, is specified to define a two-dimensional "grid" mesh which bounds and divides the envelope. From this set of grid lines, an auxiliary mesh, the "points" mesh, is created by locating points centered between grid lines and on outside boundary grid lines. It is at these points that the temperature distribution is calculated. Typical meshes are shown for the cylindrical, circular, and rectangular geometry types in Figs. 1, 2, and 3, respectively. A distinction between "internal" points and "external" points is shown in the figures. The external points represent the external coolants.

*Although the θ coordinate may extend from 0° through 360° , there is no connection between mesh points adjacent to the grid lines $\theta = 0^\circ$ and $\theta = 360^\circ$.

Fig. 1. Cylindrical geometry envelope showing grid lines and mesh points

$ZL(j)$, $ZP(j)$ = THETA DIRECTION $RL(i)$, $RP(i)$ = RADIAL DIRECTION ● = INTERNAL POINTS ▲ = EXTERNAL POINTS

Fig. 2. Circular geometry envelope showing grid lines and mesh points

In Figs. 1, 2, and 3 ZL(j) and RL(i) label the grid lines in the axial (or theta) and radial directions, respectively, and ZP(j) and RP(i) label the mesh points in the axial (or theta) and radial directions, respectively.

2.2 BLOCKS

2.2.1 General

The envelope may be subdivided into a number of blocks. Depending upon the geometry type being used, a block is a hollow or solid right circular cylinder, a polar rectangle, or a rectangle. 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 four bounding grid lines 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
	Theta	JLS	JHS
Bounding block points	Radial	ILS + 1	IHS
	Theta	JLS + 1	JHS

Figure 4 gives an example of a problem model with its enclosing envelope. The TAC2D envelope chosen for the model is not completely filled by the model because a circular 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 lines. In Fig. 4, ZL(j) and RL(i) label the grid lines 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. 4. Typical problem model in circular geometry

Table 2
INTERNAL BLOCK DEFINITION

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

Table 3
EXTERNAL BLOCK DEFINITION

Block Number (Boundary)	Block Grid Line Boundaries	
	Radial	Theta
11 (low radial)	RL(1)-RL(1)	ZL(1)-ZL(8)
12 (part high radial)	RL(8)-RL(8)	ZL(1)-ZL(4)
13 (part high radial)	RL(8)-RL(8)	ZL(4)-ZL(8)
14 (low theta)	RL(1)-RL(8)	ZL(1)-ZL(1)
15 (high theta)	RL(1)-RL(8)	ZL(8)-ZL(8)

The internal blocks are given in Table 2. The external boundaries 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 finite difference solution of the heat balance equation

$$dq = WC_p dT_c$$

where

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

W is coolant mass flow rate, lb/hr

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

T_c is coolant temperature, °F.

Coolants may flow parallel to either of the two coordinate axes. The flow direction may be either positive (in the direction of increasing coordinate value) or negative. Coolants may also flow normal to the plane of a problem which is modeled in either circular or rectangular geometry. In this case, a depth normal to the problem is specified.

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 lines, 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 adjacent to either one or both of its high radial and axial 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-9).

2.4 GAP LINES

A gap line is defined as a grid line any portion of which bounds a coolant block, or adjacent to any portion of which there is a gap. It follows that the external boundaries of the problem are always gap lines because they bound 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 TAC2D, 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 temperature 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 CONDOC 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

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

number 1. Let us say that k_r , its thermal conductivity in the radial direction, is the following function of T, its local temperature in °F.

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

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

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

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

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

3.3 FUNCTION CONTROL CONSTANTS

The need to recompile MADATA and FLODAT for each set of thermal parameter function input imposes a limitation on the use of the code to run consecutive problems. This limitation can be partially overcome through use of the constants A1 through A18, which are read in as data and are

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

included in COMMON. To illustrate, the example function of Section 3.2.2 could have been written

$$\text{RCON1}(X) = A1 - A2 * (DR - 460.0)$$

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

TAC2D 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 lines 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. 5. The code logic is described in greater detail in Appendix G.

Fig. 5. General flow of logic in TAC2D

APPENDIX A

MATHEMATICAL SOLUTION

The time dependent heat conduction differential equation is

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

where

k = thermal conductivity in each of the 2 dimensions

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

S = heat generation (Btu/ft^3-hr)

ρc = density x specific heat ($Btu/ft^3-^{\circ}R$)

τ = time (hr)

In the following discussion, Eq. (A-1) is solved by finite difference methods. Consider the envelope of the problem volume* as a set of unit cells, each containing a "mesh point" in its center. For instance, the unit cell surrounding point 0 is sketched below.

*Although the solution discussed here is two-dimensional, it is appropriate to refer to volumes because the envelope is assumed to have some depth normal to the plane of the problem.

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

$$\begin{aligned} \frac{K_1 A_1}{\Delta X_1} (T_1 - T_0) + \frac{K_2 A_2}{\Delta X_2} (T_2 - T_0) + \frac{K_3 A_3}{\Delta X_3} (T_3 - T_0) \\ + \frac{K_4 A_4}{\Delta X_4} (T_4 - T_0) + S_0 V_0 = \frac{(T'_0 - T_0) \rho c V_0}{\Delta\tau} \quad (A-2) \end{aligned}$$

where

- A_i = average cell surface area between Point 0 and i ($i = 1, 4$)
- ΔX_i = distance between Points 0 and i
- K_i = composite thermal conductivity between Points 0 and i in the principal coordinate directions
- T_i = temperature of Point i ($^{\circ}R$)
- T'_0 = temperature of Point 0 ($^{\circ}R$) 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 Peaceman and Rachford 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 Peaceman and Rachford is presented below. To simplify equations like Eq. (A-2), we replace $K_i A_i/\Delta X_i$ with C_i ,

$S_0 V_0$ with W , and $(\rho c V_0 / \Delta \tau)$ with C_5 . Then

$$\begin{aligned} & C_1(T_1 - T_0) + C_2(T_2 - T_0) \\ & + C_3(T_3 - T_0) + C_4(T_4 - T_0) = C_5(T'_0 - T_0) - W. \end{aligned} \quad (A-3)$$

Define an "old" temperature as a known value of T at the beginning of a time increment $\Delta \tau$. Old temperatures will be denoted by an unprimed T . New temperatures will be denoted by a primed T . By successively replacing the old temperatures with new temperatures in sets of T' and T'' , we may restate Eq. (A-3) in the two different forms of Eqs. (A-4) and (A-5).

$$\begin{aligned} & C_1 \left(\frac{(T'_1 + T_1)}{2} - T_0 \right) + C_2 \left(\frac{(T'_2 + T_2)}{2} - T_0 \right) \\ & + C_3(T_3 - T_0) + C_4(T_4 - T_0) = C_5(T'_0 - T_0) - W \end{aligned} \quad (A-4)$$

and

$$\begin{aligned} & C_1 \left(\frac{(T'_1 + T_1)}{2} - T_0 \right) + C_2 \left(\frac{(T'_2 + T_2)}{2} - T_0 \right) \\ & + C_3 \left(\frac{(T''_3 + T_3)}{2} - T_0 \right) + C_4 \left(\frac{(T''_4 + T_4)}{2} - T_0 \right) = C_5(T''_0 - T_0) - W \end{aligned} \quad (A-5)$$

The difference between Eqs. (A-5) and (A-4) is

$$C_3 \frac{(T''_3 - T_3)}{2} + C_4 \frac{(T''_4 - T_4)}{2} = C_5(T''_0 - T'_0) \quad (A-6)$$

Because it is desired to solve the equations line by line in both coordinate directions, we rewrite Eqs. (A-4) and (A-6) as

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

or

$$AT'_1 + BT'_0 + CT'_2 = D, \quad (A-7)$$

$$\frac{C_3 T_3''}{2} - C_5 T_0'' + \frac{C_4 T_4''}{2} = -C_5 T_0' + \frac{C_3 T_3}{2} + \frac{C_4 T_4}{2}$$

or

$$AT_3'' + BT_0'' + CT_4'' = D \quad . \quad (A-8)$$

A set of N equations of the form

$$A_i T_{i-1} + B_i T_i + C_i T_{i+1} = D_i \quad (A-12)$$

may be solved as outlined below.

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 $T_i = Y_i$, $T_i = Y_i - X_i T_{i+1}$. The temperatures thus calculated in one coordinate direction become the T_0' for the other coordinate direction.

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 TAC2D are derived here in the nomenclature of the code.

In TAC2D, Point 0 is a point (I,J) whose unit cell is bounded by the following grid lines:

Radial I-1, I

Axial J-1, J

For each unit cell, two coordinate-oriented constants and a volume are calculated. In the radial direction,

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

and

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

In the axial direction,

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

and

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

For the volume,

$$V_0 = V(I,J) \quad (B-3)$$

In order to define the area components (i.e., A_1 through A_4) of these constants in circular or rectangular geometries, some depth normal to the plane of the calculation must be assigned. This depth, HEIGHT, has a value of one foot unless it has been assigned some other value through the use of normal flow coolants (see Section 2.2.2). The remaining nomenclature to be used in the two coordinate directions is as follows:

	<u>Grid Line</u>	<u>Point</u>	<u>Gap Thickness</u>
Radial (or X)	RL	RP	Δg_R
Axial (or theta or Y)	ZL	ZP	Δg_Z

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. 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 each of the three geometry types. For the cylindrical geometry,

$$RR(I,J) = \left[\frac{2\pi}{\ln\left(\frac{RP(I+1)}{RP(I)}\right)} \right] \left[ZL(J) - ZL(J-1) - \frac{\Delta g_Z(I) + \Delta g_Z(I+1)}{2.0} \right] \quad (B-4)$$

and

$$RZ(I,J) = \frac{\pi \left\{ \left[RL(I) - \frac{\Delta g_R(J) + \Delta g_R(J+1)}{2.0} \right]^2 - \left[RL(I-1) \right]^2 \right\}}{ZP(J+1) - ZP(J)} \quad (B-5)$$

and

$$V(I,J) = \pi \left\{ \left[RL(I) - \Delta g_R(J) \right]^2 - \left[RL(I-1) \right]^2 \right\} \left\{ ZL(J) - ZL(J-1) - \Delta g_Z(I) \right\} \quad (B-6)$$

For the circular geometry

$$RR(I,J) = \left[\frac{HEIGHT}{\ln\left(\frac{RP(I+1)}{RP(I)}\right)} \right] \left[ZL(J) - ZL(J-1) - \frac{\Delta g_Z(I) + \Delta g_Z(I+1)}{2.0} \right] \quad (B-7)$$

and

$$RZ(I,J) = \left\{ \frac{HEIGHT}{RP(I)[ZP(J+1) - ZP(J)]} \right\} \left\{ RL(I) - RL(I-1) - \frac{\Delta g_R(J) + \Delta g_R(J+1)}{2.0} \right\} \quad (B-8)$$

and

$$V(I,J) = \left\{ \frac{HEIGHT}{2.0} \right\} \left\{ \left[RL(I) - \Delta g_R(J) \right]^2 - \left[RL(I-1) \right]^2 \right\} \left\{ ZL(J) - ZL(J-1) - \Delta g_Z(I) \right\} \quad (B-9)$$

For the rectangular geometry

$$RR(I,J) = \left[\frac{HEIGHT}{RP(I+1) - RP(I)} \right] \left[ZL(J) - ZL(J-1) - \frac{\Delta g_Z(I) + \Delta g_Z(I+1)}{2.0} \right] \quad (B-10)$$

and

$$RZ(I,J) = \left[\frac{HEIGHT}{ZP(J+1) - ZP(J)} \right] \left[RL(I) - RL(I-1) - \frac{\Delta g_R(J) + \Delta g_R(J+1)}{2.0} \right] \quad (B-11)$$

and

$$V(I,J) = [HEIGHT] \left[RL(I) - RL(I-1) - \Delta g_R(J) \right] \left[ZL(J) - ZL(J-1) - \Delta g_Z(I) \right] \quad (B-12)$$

APPENDIX C

COMPOSITE CONDUCTIVITIES

Depending upon the part of the TAC2D 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 TAC2D in determining what calculations are to be performed.

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.

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

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

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

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

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

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

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

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

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

Now, the equation for K may be rewritten as

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

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

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

and, in the radial direction for rectangular geometry,

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

In the theta direction for circular geometry,

$$\begin{aligned}
 A &= A_1 = A_2 \\
 \Delta X &= [ZP(J + 1) - ZP(J)][RP(I)] \\
 \Delta X_1 &= [ZL(J) - ZP(J)][RP(I)] \quad ,
 \end{aligned}
 \tag{C-7}$$

and, in the axial direction for cylindrical and rectangular geometry,

$$\begin{aligned}
 A &= A_1 = A_2 \\
 \Delta X &= ZP(J + 1) - ZP(J) \\
 \Delta X_1 &= ZL(J) - ZP(J) \quad .
 \end{aligned}
 \tag{C-8}$$

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

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

Geometry Type	Name (Coordinate)	
	RLN(I)	ZLN(J)
Cylindrical	$\frac{\ln\{[RL(I)]/[RP(I)]\}}{\ln\{[RP(I + 1)]/[RP(I)]\}}$	$\frac{ZL(J) - ZP(J)}{ZP(J + 1) - ZP(J)}$
Circular	$\frac{\ln\{[RL(I)]/[RP(I)]\}}{\ln\{[RP(I + 1)]/[RP(I)]\}}$	$\frac{ZL(J) - ZP(J)}{ZP(J + 1) - ZP(J)}$
Rectangular	$\frac{RL(I) - RP(I)}{RP(I + 1) - RP(I)}$	$\frac{ZL(J) - ZP(J)}{ZP(J + 1) - ZP(J)}$

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

$$\begin{aligned} \frac{KA(T_1 - T_2)}{\Delta X} &\equiv \frac{k_1 A_1 (T_1 - T'')}{\Delta X_1} = \frac{k_2 A_2 (T' - T_2)}{\Delta X_2} \\ &= \frac{k_g A_g (T'' - T')}{\Delta g} + \frac{\sigma A_g (T''^4 - T'^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \end{aligned} \quad (C-9)$$

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

C.3.1 Iterative Evaluation of Gap Surface Temperatures, T' and T''

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

$$\frac{k_1 A_1}{\Delta X_1} (T_1 - T'') = \frac{k_2 A_2}{\Delta X_2} (T' - T_2) \quad (C-10)$$

When solving for T'' , one obtains

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

$$\frac{k_2 A_2}{\Delta X_2} (T' - T_2) = \frac{k A_g}{\Delta g} (T'' - T') \left[1 + \frac{\sigma \Delta g (T''^2 + T'^2) (T'' + T')}{k_g \left(\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1 \right)} \right] \quad (C-12)$$

and

$$T' - T_2 = \frac{k A_g \Delta X_2}{k_2 A_2 \Delta g} (T'' - T') \quad G \equiv H(T'' - T') \quad (C-13)$$

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

$$T' - T_2 = H \left[B(T_2 - T') + T_1 - T' \right] \quad (C-14)$$

and

$$T' = \frac{T_2(HB + 1) + HT_1}{HB + 1 + H} \quad (C-15)$$

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 TAC2D, this area is determined at the outside of the gap. In other words, if in the preceding figure the coordinate X increases to the right, then A_g is assumed to be the same as A . The other inaccuracy is introduced in the iterative method of obtaining T' ; however, this error can be made as small as desired. Presently, the iterations are stopped when two consecutive values of T' are within 1°F of each other. Up to ten iterations are allowed to accomplish this.

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

C.3.2 Geometrical Constants for Gaps

Three geometrical terms — $(A_2/\Delta X_2)/(A/\Delta X)$, $(A_2/\Delta X_2)/(A_1/\Delta X_1)$, and $A_g/(A_2/\Delta X_2)$ — were required for the equations that determine gap surface temperatures and composite conductivity with a gap present. Considerations for the determination of these terms are given below.

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

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

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

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

$$A_g = RL(I) ,$$

and

$$\Delta g = \Delta g_R . \quad (C-16)$$

In the radial direction for rectangular geometry,

$$A = A_1 = A_2 = A_g ,$$

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

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

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

and

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

In the axial direction for rectangular and cylindrical geometries,

$$A = A_1 = A_2 = A_g ,$$

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

and

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

In the theta direction for circular geometry,

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

and

$$\Delta g = [RP(I)] [\Delta g_Z] \quad (C-19)$$

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 three available geometry types are given in Tables C.2, C.3, and C.4.

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

Geometry Type	Name (Gap Line) (See Note 1)	
	RATIOK(IG)	ZATIOK(JG)
Cylindrical	$\frac{\ln \{ [RP(I + 1)] / [RP(I)] \}}{\ln \{ [RP(I + 1)] / [RL(I)] \}}$	$\frac{ZP(J + 1) - ZP(J)}{ZP(J + 1) - ZL(J)}$
Circular	$\frac{\ln \{ [RP(I + 1)] / [RP(I)] \}}{\ln \{ [RP(I + 1)] / [RL(I)] \}}$	$\frac{ZP(J + 1) - ZP(J)}{ZP(J + 1) - ZL(J)}$
Rectangular	$\frac{RP(I + 1) - RP(I)}{RP(I + 1) - RL(I)}$	$\frac{ZP(J + 1) - ZP(J)}{ZP(J + 1) - ZL(J)}$

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

Geometry Type	Name (Gap Line Index, Point Index) (See Note 2)	
	RATIOB(IG,J)	ZATIOB(JG,I)
Cylindrical	$\ln \left[\frac{RL(IHS) - RDG(L)}{RP(IHS)} \right] / \ln \left[\frac{RP(IHS+1)}{RL(IHS)} \right]$	$\frac{ZL(JHS) - ZP(JHS) - ZDG(L)}{ZP(JHS+1) - ZL(JHS)}$
Circular	$\ln \left[\frac{RL(IHS) - RDG(L)}{RP(IHS)} \right] / \ln \left[\frac{RP(IHS+1)}{RL(IHS)} \right]$	$\frac{ZL(JHS) - ZP(JHS) - ZDG(L)}{ZP(JHS+1) - ZL(JHS)}$
Rectangular	$\frac{RL(IHS) - RP(IHS) - RDG(L)}{RP(IHS+1) - RL(IHS)}$	$\frac{ZL(JHS) - ZP(JHS) - ZDG(L)}{ZP(JHS+1) - ZL(JHS)}$

TABLE C.4
GEOMETRICAL CONSTANT
 $Ag/(A_2/\Delta X_2)$

Geometry Type	Name (Gap Line Index) (See Note 1)	
	RATIOH(IG)	ZATIOH(IG)
Cylindrical	$RL(I) \left[\ln \frac{RP(I+1)}{RL(I)} \right]$	$ZP(J+1) - ZL(J)$
Circular	$RL(I) \left[\ln \frac{RP(I+1)}{RL(I)} \right]$	$ZP(J+1) - ZL(J)$
Rectangular	$RP(I+1) - RL(I)$	$ZP(J+1) - ZL(J)$

NOTES:

1. IG and JG are the gap line indices at radial grid line I and axial grid line J, respectively.
2. IHS and JHS are the grid line indices of the high radial and high axial boundaries, respectively, of Block L. RDG(L) and ZDG(L) are the gap widths, respectively, at IHS and JHS.

C.4 COOLANTS

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

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

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

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

C.4.1 Coolant on High Index Side of Grid Line

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

thermal conductivity K_1 between the mesh point and the fluid is

$$\begin{aligned} \frac{k_1 A_1^*}{\Delta X_1} (T_1 - T_{s1}) &= h_1 A_1 (T_{s1} - T_c) + \frac{\sigma A_1 (T_{s1}^4 - T_{s2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \\ &\equiv \frac{K_1 A_1^{**} (T_1 - T_c)}{\Delta X} , \end{aligned} \quad (C-20)$$

where

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

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

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

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

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

ϵ_1, ϵ_2 = thermal emissivities

The term $h_1 A_1 (T_{s1} - T_c)$ represents the heat transfer by convection to the fluid, and

$$\frac{\sigma A_1 (T_{s1}^4 - T_{s2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \quad (C-21)$$

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

A composite heat transfer coefficient H_1 is defined by

$$h_1 A_1 (T_{s1} - T_c) + \frac{\sigma A_1 (T_{s1}^4 - T_{s2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \equiv H_1 A_1 (T_{s1} - T_c) . \quad (C-22)$$

Let

$$\frac{\sigma \left(T_{s1}^4 - T_{s2}^4 \right)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} = \text{RAD} \quad . \quad (\text{C-23})$$

Now, the surface temperature T_{s1} is calculated as

$$\frac{k_1 A_1^*}{\Delta X_1^*} (T_1 - T_{s1}) = H_1 A_1 (T_{s1} - T_c) \quad , \quad (\text{C-24})$$

and

$$T_{s1} = \frac{H_1 A_1 T_c + \frac{k_1 A_1^*}{\Delta X_1^*} T_1}{H_1 A_1 + \frac{k_1 A_1^*}{\Delta X_1^*}} = \frac{\frac{1}{k_1} T_c + \frac{1}{H_1} \frac{A_1^* / \Delta X_1^*}{A_1} T_1}{\frac{1}{k_1} + \frac{1}{H_1} \frac{A_1^* / \Delta X_1^*}{A_1}} \quad . \quad (\text{C-25})$$

By substitution of T_{s1} in Eq. (C-22) one may solve for H_1 as

$$\frac{1}{H_1} = \frac{T_1 - T_c - \frac{\text{RAD}}{k_1} \frac{A_1}{A_1^* / \Delta X_1^*}}{h_1 \left(T_1 - T_c + \frac{\text{RAD}}{h_1} \right)} \quad . \quad (\text{C-26})$$

Rewriting (C-20) as

$$\frac{K_1 A_1^{**} (T_1 - T_c)}{\Delta X} = H_1 A_1 (T_{s1} - T_c) \quad , \quad (\text{C-27})$$

one may solve for K_1 by substitution of (C-25) in (C-27):

$$K_1 = \frac{1}{\frac{1}{k_1} \frac{A_1^{**} / \Delta X}{A_1^* / \Delta X_1^*} + \frac{1}{H_1} \frac{A_1^{**} / \Delta X}{A_1}} \quad (\text{C-28})$$

C.4.2 Coolant on Low Index Side of Grid Line

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

$$\begin{aligned} \frac{K_2 A_2^*}{\Delta X_2^*} (T_2 - T_{s_2}) &= h_2 A_2 (T_{s_2} - T_c) + \frac{\sigma A_1 (T_{s_2}^4 - T_{s_1}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} \\ &= \frac{K_2 A_2^{**} (T_2 - T_c)}{\Delta X} \end{aligned} \quad (C-29)$$

where

- A_2 = area across which heat is transferred at the coolant surface
- A_2^* = average area across which heat is transferred in the material
- A_2^{**} = average area for heat transfer between T_2 and T_c
- ΔX_2^* = distance across which heat travels to reach the coolant surface

A composite heat transfer coefficient H_2 is defined by

$$h_2 A_2 (T_{s_2} - T_c) - \frac{\sigma A_1 (T_{s_1}^4 - T_{s_2}^4)}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} = H_2 A_2 (T_{s_2} - T_c) \quad (C-30)$$

When one follows the same scheme as with the coolant on the high index side of the grid line, the surface temperature T_{s_2} , the composite heat transfer coefficient H_2 and the composite thermal conductivity K_2 are obtained as

$$T_{s_2} = \frac{\frac{1}{k_2} T_c + \frac{1}{H_2} \frac{A_2^* / \Delta X_2^*}{A_2} T_2}{\frac{1}{k_2} + \frac{1}{H_2} \frac{A_2^* / \Delta X_2^*}{A_2}}, \quad (C-31)$$

and

$$\frac{1}{H_2} = \frac{T_2 - T_c + \frac{RAD}{k_2} \frac{A_1}{A_2} \frac{A_2}{A_2^*/\Delta X_2^*}}{h_2 \left(T_2 - T_c - \frac{RAD}{h_2} \frac{A_1}{A_2} \right)} \quad (C-32)$$

and

$$K_2 = \frac{1}{\frac{1}{k_2} \frac{A_2^{**}/\Delta X}{A_2^*/\Delta X_2^*} + \frac{1}{H_2} \frac{A_2^{**}/\Delta X}{A_2}} \quad (C-33)$$

C.4.3 Geometrical Constants for Coolants

Some of the geometrical constants appearing in Eqs. (C-25) through (C-33) are given in Tables C.5 through C.8. The nomenclature used is that

Table C.5
GEOMETRICAL CONSTANT
 $(A_1^*/\Delta X_1^*)/A_1$

Geometry Type	Coordinate Direction	
	Radial	Axial
Cylindrical	$\frac{1.0}{[RP(I + 1) - RP(I)] RLN(I)}$	$\frac{1.0}{ZL(J) - ZP(J)}$
Circular	$\frac{1.0}{[RP(I + 1) - RP(I)] RLN(I)}$	$\frac{1.0}{[ZL(J) - ZP(J)] [RP(I)]}$
Rectangular	$\frac{1.0}{[RP(I + 1) - RP(I)] RLN(I)}$	$\frac{1.0}{ZL(J) - ZP(J)}$

TABLE C.6
GEOMETRICAL CONSTANT

$$(A_2^*/\Delta X_2^*)/A_2$$

Geometry Type	Coordinate Direction	
	Radial	Axial
Cylindrical	$\frac{\text{RATIOK(IG)}}{\text{RP(I + 1)} - \text{RP(I)}}$	$\frac{1.0}{\text{ZP(J + 1)} - \text{ZL(J)}}$
Circular	$\frac{\text{RATIOK(IG)}}{\text{RP(I + 1)} - \text{RP(I)}}$	$\frac{1.0}{[\text{ZP(J + 1)} - \text{ZL(J)}] [\text{RP(I)}]}$
Rectangular	$\frac{\text{RATIOK(IG)}}{\text{RP(I + 1)} - \text{RP(I)}}$	$\frac{1.0}{\text{ZP(J + 1)} - \text{ZL(J)}}$

TABLE C.7
GEOMETRICAL CONSTANT
 $(A_1^{**}/\Delta X)/A_1$ or $(A_2^{**}/\Delta X)/A_2$

Geometry Type	Coordinate Direction	
	Radial	Axial
Cylindrical	$\frac{1.0}{\text{RL(I)} \left[\ln \frac{\text{RP(I + 1)}}{\text{RP(I)}} \right]}$	$\frac{1.0}{\text{ZP(J + 1)} - \text{ZP(J)}}$
Circular	$\frac{1.0}{\text{RL(I)} \left[\ln \frac{\text{RP(I + 1)}}{\text{RP(I)}} \right]}$	$\frac{1.0}{[\text{ZP(J + 1)} - \text{ZP(J)}] [\text{RP(I)}]}$
Rectangular	$\frac{1.0}{\text{RP(I + 1)} - \text{RP(I)}}$	$\frac{1.0}{\text{ZP(J + 1)} - \text{ZP(J)}}$

TABLE C. 8
GEOMETRICAL CONSTANT
 A_1/A_2

Geometry Type	Coordinate Direction (See Note Below)	
	Radial	Axial
Cylindrical	$\frac{RL(ILS - 1)}{RL(IHS)}$	1.0
Circular	$\frac{RL(ILS - 1)}{RL(IHS)}$	1.0
Rectangular	1.0	1.0

NOTE:

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 coolant boundaries was considered in Appendix C. In this appendix, the coolant temperature calculation is discussed. With respect to the TAC2D code, the following scheme is used in handling the coolants:

1. The solution for mesh point temperatures is carried out as described in Appendix A. During this solution, the points in coolants experience no change in temperature. This is accomplished by setting the thermal capacitance of all coolant nodes at 1.0×10^{12} Btu/°R. The effect of radiation and convection at coolant boundaries on mesh point temperatures adjacent coolants is accounted for in the solution through use of the composite conductivities described in Section C.4 of Appendix C.
2. New coolant temperatures based on the results of the above solution are calculated.
3. New coolant block boundary temperatures based on the results of Steps 2 and 3 above are calculated. These are used to determine a new value for the radiation transport as defined in Eq. (C-23) of Appendix C.
4. Composite thermal conductivities between the bounding material and the coolant (Eqs. (C-28) and (C-33) of Appendix C) are calculated, and another iteration is performed.

In TAC2D, 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

$$WC_p dF - \sum_i h_i S_i (T_i - F) dz = 0, \quad (D-1)$$

where

dF = differential coolant temperature increment

h_i = heat transfer coefficient (Btu/ft²-hr-°R)

S_i = linear surface along coolant boundary associated with one point

dz = differential height of increment along flow direction

T_i = surface temperature

Rearrangement gives

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

When

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

are defined, the equation becomes

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

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

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

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

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

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

and

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

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

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

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

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

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

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

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

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

When

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

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

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

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

and

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

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. TAC2D contains a steady-state option which, for most problems, has been found to circumvent these difficulties when only a steady-state solution is desired. All calculations are performed in the normal manner except that control of the pseudo-transient is transferred to a special subroutine which assigns the capacitance terms and time steps by the method described below.

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

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

where

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

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

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

For an implicit method of solution such as that used in TAC2D, $\Delta\tau_i$ has been found through experience to be proportional to the right hand side of Eq. (E-1). Therefore, at the limit of stability,

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

where the β_i are unknown constants of proportionality.

In TAC2D, 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

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

Eq. (E-2) becomes:

$$C_i = \sum_{j=1}^4 K_{ji} \quad (E-3)$$

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 five hundred hours according to the function

$$\Delta\tau' = \left(\frac{N}{10}\right)^3 + N + 1 \quad . \quad (E-4)$$

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

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

then the current value of $\Delta\tau'$ is halved 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 conditions 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

$$R_i = \frac{\left| \frac{C_i(T_i - T'_i)}{\Delta\tau i'} \right|}{\sum_{j=1}^4 \left| K_{ji} \left(\frac{(T_i + T'_i)}{2} - \frac{(T_j + T'_j)}{2} \right) \right| + |W_i|} \quad (E-5)$$

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.*

*These values are retained on an external storage device.

The R_i are calculated after every tenth iteration. The convergence criterion which must be satisfied is

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

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.

APPENDIX F

SAMPLE PROBLEM

A sample problem is presented to compare the results from TAC2D with an analytical solution. The objective is to show that the TAC2D program yields the correct solution.

F.1 PROBLEM DESCRIPTION

The problem is to determine the transient temperature distribution in a right circular cylinder which is initially at temperature T_1 . At time $t = 0$, the temperature at the surface is instantaneously changed to T_2 and maintained at that value. Mathematically the problem is defined by the following equations:

$$\frac{1}{r} \frac{\partial}{\partial r} r \frac{\partial T}{\partial t} + \frac{\partial^2 T}{\partial z^2} = \frac{1}{\kappa} \frac{\partial T}{\partial t} \quad ; \quad 0 < r < R \quad (F-1)$$

$$T(r, z, 0) = T_1 \quad (F-2)$$

$$T(R, z, t) = T_2 \quad (F-3)$$

$$T(r, \pm \frac{L}{2}, t) = T_2 \quad (F-4)$$

where t is the time, r is the radius, z the axial coordinate, R the outside radius of the cylinder, L the length of the cylinder and κ the diffusivity. Further,

$$\kappa = \frac{k}{\rho c} \quad (F-5)$$

where k is the thermal conductivity, ρ the density and c the specific heat capacity. For the specific problem analyzed, the following numerical values were used:

$$R = 12.0 \text{ in.}; \quad L = 48.0 \text{ in.}$$

$$k = 20.0 \text{ Btu/hr-ft-}^\circ\text{F}; \quad \rho c = 40.0 \text{ Btu/ft}^3\text{-}^\circ\text{F}$$

$$T_1 = 0.0 \text{ }^\circ\text{F}; \quad T_2 = 1000.0^\circ\text{F}$$

F.2 ANALYTICAL SOLUTION

It may be shown (Ref. 6) that the solution is

$$\frac{T - T_1}{T_2 - T_1} = 1 - f(z,t) g(r,t) \quad (\text{F-6})$$

$$f(z,t) = \frac{4}{\pi} \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{(2n-1)} \cos \left[(2n-1) \frac{\pi z}{L} \right] e^{-\kappa \left(\frac{(2n-1)\pi}{L} \right)^2 t} \quad (\text{F-7})$$

$$g(r,t) = 2 \sum_{n=1}^{\infty} \frac{J_0 \left(\frac{r}{R} \right) \gamma_n}{\gamma_n J_1(\gamma_n)} e^{-\kappa \gamma_n^2 t} \quad (\text{F-8})$$

where the γ_n are the roots of

$$J_0(\gamma_n) = 0 \quad (\text{F-9})$$

The roots γ_n of Eq. (F-9) and the functions $J_1(\gamma_n)$ are tabulated in Ref. 7 and need not be computed.

From the definition of the problem there is symmetry about the geometric center of the cylinder and the origin of the coordinate system taken at that point, as is reflected in the boundary conditions, Eqs. (F-3) and (F-4).

F.3 NUMERICAL SOLUTION WITH TAC2D

A cross section of the problem model for TAC2D is shown in Fig. F.1. The model extends only to the axial midplane of the cylinder where an adiabatic boundary may be specified by virtue of the symmetry condition described in Section F.2 above. The solid material is represented by one material block. The boundary conditions on the four external boundaries are described by Coolants 1 through 4 (specifically, Coolant Blocks 1 through 4). The material and coolant thermal parameters, as specified by the input functions, are given in Table F.1. All coolants have the standard specific

Fig. F.1. TAC2D sample problem, thermal model

TABLE F.1
INPUT THERMAL PARAMETER FUNCTIONS FOR TAC2D SAMPLE PROBLEM

C MATERIAL THERMAL PARAMETERS

SPEC1(X)=40.0

RCON1(X)=20.0

ACON1(X)=20.0

C COOLANT THERMAL PARAMETERS

H3A(X)=1.0E+08

FLO3A(X)=1.0E+08

TIN3A(X)=1460.0

H4A(X)=1.0E+08

FLO4A(X)=1.0E+08

TIN4A(X)=1460.0

heat of 1.0 Btu/lb-°F. Coolants 1 and 2, which represent the adiabatic external boundaries, have the standard heat transfer coefficient of 10^{-6} Btu/hr-ft²-°F and the standard flow rate of 10^6 lb/hr. A portion of the TAC2D printout information, Table F.2, shows the block and grid line specifications, as well as time history information to run to 2.0 hours.

F.4 COMPARISON OF THE TAC2D SOLUTION WITH THE ANALYTICAL SOLUTION

A comparison of the output from the code with the series solution is shown in Fig. F.2 where the temperature versus time function is plotted at three representative points within the cylinder. It can be seen that the results from TAC2D are almost identical to the series solution results. The maximum difference between the two sets of results is about 2°F out of a mean magnitude of 100°F.

TABLE F.2

EXCERPTS FROM TAC2D SAMPLE PROBLEM OUTPUT SHOWING
BLOCK, GRID PLANE AND TIME HISTORY SPECIFICATIONS

BLOCK NUMBER	BLOCK DESCRIPTION								
	BOUNDARIES					GAPS			
	LOW RADIAL (INCHES)	HIGH RADIAL (INCHES)	LOW AXIAL (INCHES)	HIGH AXIAL (INCHES)	MATERIAL	RADIAL (INCHES)	MATERIAL	AXIAL (INCHES)	MATERIAL
1	.0000	12.0000	.0000	24.0000	1				
2	.0000	12.0000	.0000	.0000	-1				
3	.0000	.0000	.0000	24.0000	-2				
4	.0000	12.0000	24.0000	24.0000	-3				
5	12.0000	12.0000	.0000	24.0000	-4				

TABLE F.2
(Continued)

RADIAL BOUNDARY ASSIGNMENTS			
SEQUENCE NUMBER	POINT LOCATION (INCHES)	GRID LINE LOCATION (INCHES)	COOLANT OR GAP BOUNDARY NUMBER
1	.0000	.	
2	.5000	.0000	1
3	1.5000	1.0000	
4	2.5000	2.0000	
5	3.5000	3.0000	
6	4.5000	4.0000	
7	5.5000	5.0000	
8	6.5000	6.0000	
9	7.5000	7.0000	
10	8.5000	8.0000	
11	9.5000	9.0000	
12	10.5000	10.0000	
13	11.5000	11.0000	
14	12.0000	12.0000	2

TABLE F.2
(Continued)

AXIAL BOUNDARY ASSIGNMENTS			
SEQUENCE NUMBER	POINT LOCATION (INCHES)	GRID LINE LOCATION (INCHES)	COOLANT OR GAP BOUNDARY NUMBER
1	.0000		
2	.5000		
3	1.5000		
4	2.5000		
5	3.5000		
6	4.5000		
7	5.5000		
8	6.5000		
9	7.5000		
10	8.5000		
11	9.5000		
12	10.5000		
13	11.5000		
14	12.5000		
15	13.5000		
16	14.5000		
17	15.5000		
18	16.5000		
19	17.5000		
20	18.5000		

1

TABLE F.2
(Continued)

Axial Boundary Assignments (Continued)

20			19.0000
21	20	19.5000	20.0000
22	21	20.5000	21.0000
23	22	21.5000	22.0000
24	23	22.5000	23.0000
25	24	23.5000	24.0000
26	25	24.0000	

2

64

TIME HISTORY

END OF TIME PERIOD			TIME STEP			PRINT FREQUENCY
HOURS	MINUTES	SECONDS	HOURS	MINUTES	SECONDS	
5.000000-01	3.000000+01	1.800000+03	1.000000-02	6.000000-01	3.600000+01	1
1.000000+00	6.000000+01	3.600000+03	1.000000-02	6.000000-01	3.600000+01	5
2.000000+00	1.200000+02	7.200000+03	1.000000-02	6.000000-01	3.600000+01	10

Fig. F.2. Transient temperatures in a right circular cylinder - comparison of TAC2D results with series solution

APPENDIX G

CODE ORGANIZATION AND FUNCTIONS OF SUBROUTINES IN TAC2D

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 TAC2D

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 lines; 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 lines and block boundaries. Identifies any material, coolant or gas numbers which are not allowed.

called by MP1

calls ERROR

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

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 line.

called by MP1

calls BOUNDA, ERROR

- 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 lines; relates the gap line indices to the grid line 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-4) through (B-12); prints all values of the above quantities if requested through input option.

called by MP1

calls ERROR, ARRAY

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

called by GEOMET, PRETEM,
PRINT, STEADY

calls None

6. Subroutine DPRINT - Prints the locations and indices of the points, grid lines, and gap lines.

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 specifies 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, ARRAY

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

called by MP1

calls None

B. Subroutine MP2 - Main program of Computational Section

called by MAIN

calls ERROR2
STEADY*
BLOCK
CONDUCT
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, CONDUCT,
COOL, STEP, SURT,
PRINT, ARRAY, PUN

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

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, CONDUCT

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 Functions which define coolant thermal parameters as described in Section 3.2

4. Subroutine CONDUCT - 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 CUSTOM - Performs any additional computations and printing according to FORTRAN V instructions supplied by the code user.

called by PRINT

calls As programmed by the code user

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

TAC2D CODE LISTING

The following pages are a card image listing of the TAC2D code. The elements are arranged in the calling sequence outlined in Appendix G. They are page indexed in alphabetical order below so that individual elements may be easily located.

<u>Element Name</u>	<u>Page</u>
ARRAY	118
BLOCK	146
BOUNDA	109
CDE (PDP procedure element)	78
CHECK	102
CONDOC	176
CONSTA	112
COOL	186
COOL1 through COOL15	174
CUSTOM	211
DPRINT	120
ERROR	100
ERROR2	139
EXTRA	132
FCARD	101
FLOWCA	125
FLODAT	160
FMAT1 through FMAT15	158
GEOMET	115
INITEM	121
INPUT	91
MAIN	80

<u>Element Name</u>	<u>Page</u>
MADATA	149
MP1	88
MP2	136
POINTS	104
PRETEM	133
PRINT	201
PUN	212
STEADY	140
STEP	184
SURT	194
TAC2D (processed map)	77
TIME	129

TAC2D (processed map)

```
      SEG  MAIN=*(MP1,MP2L)
MP2L  SEG  MP2-STEADY-PUN=*(LNK1,LNK2,LNK3)
LNK1  SEG  COUL-STEP-CONduc-BLOCK-SURT
LNK2  SEG  PRINT
LNK3  SEG  ERROR2
```


	2	ST	,TOUT	,TIN		COMD 460
C		COMMON /XMAX /	MAXFLO	,MMAX	,MAXRP	,MAXZP
	1	MAXNB	,MAXMAT	,MAXRG	,MAXZG	
						COMD 470
						COMD 480
C		COMMON /XSUBS /	IM	,IM1	,IMAX	,IGLS
	1	IGHS	,NRG	,LMAX	,JM	
	2	JM1	,JMAX	,JGLS	,JGHS	
	3	NZG				
						COMD 490
						COMD 500
						COMD 510
						COMD 520
						COMD 530
						COMD 540
C		COMMON /MISCXX/	CURTI	,FIRST	,GAS	,ICOUNT
	1	ISHAPE	,ITER	,NPRINT	,SCALE	
	2	NI	,TB	,TH	,DATI	
	3	HEIGHT	,NC	,IERROR(2)	,ZA (14)	
	4	PNAME (12)	,CS1	,CS2	,CS3	
						COMD 550
						COMD 560
						COMD 570
						COMD 580
						COMD 590
						COMD 600
C		COMMON /XMAXAX/	ILS	,IHS	,JLS	,JHS
						COMD 610
						COMD 620
C		COMMON /XTRA /	A1	,A2	,A3	,A4
	1	A5	,A6	,A7	,A8	
	2	A9	,A10	,A11	,A12	
	3	A13	,A14	,A15	,A16	
	4	A17	,A18			
						COMD 630
						COMD 640
						COMD 650
						COMD 660
						COMD 670
						COMD 680
C		COMMON /IGXJX /	SAPR (1GQ,JQ),		MATRG (1GQ,JQ),	
	1	RBBTL (1GQ,JQ),			RBBTH (1GQ,JQ),	
	2	REML (1GQ,JQ),			REMH (1GQ,JQ),	
	3	RATIOB(1GQ,JQ)				
						COMD 690
						COMD 700
						COMD 710
						COMD 720
						COMD 730
						COMD 740
C		COMMON /JGXJX /	SAPZ (JGQ,IQ),		MATZG (JGQ,IQ),	
	1	ZBBTL (JGQ,IQ),			ZBBTH (JGQ,IQ),	
	2	ZEML (JGQ,IQ),			ZEMH (JGQ,IQ),	
	3	ZATIOB(JGQ,IQ)				
						COMD 750
						COMD 760
						COMD 770
						COMD 780
						COMD 790
C		COMMON /IXJX /	RR (IQ,JQ),		RZ (IQ,JQ),	
	1	V (IQ,JQ),			W (IQ,JQ),	
	2	CONR (IQ,JQ),			CONZ (IQ,JQ),	
	3	RCP (IQ,JQ),			TT (IQ,JQ),	
	4	T (IQ,JQ)				
						COMD 800
						COMD 810
						COMD 820
						COMD 830
						COMD 840
C		EQUIVALENCE	(MT ,W),(RBBTL ,RBL),(RBBTH ,RBH),			
	1	(REML ,RDG),(REMH ,MGR),(ZBBTL ,ZBL),				COMD 850
	2	(ZBBTH ,ZBH),(ZEM ,ZDG),(ZEMH ,MGZ),				COMD 860
	3	(CONR ,XR),(CONZ ,KZ),(CONR ,CARD),				COMD 870
	4	(FLUDEP,NFLO)				COMD 880
						COMD 890
						COMD 900
C=		=====				COMD 910
C=		=====				COMD 920
		END				COMD 930

```

C= =====MAIN 10
C INCLUDE COMDIM MAIN 20
C =====MAIN 30
C MAIN PROGRAM OF IAC2D MAIN 40
C =====MAIN 50
C =====MAIN 60
C PROGRAM VARIABLE DEFINITION (INITIAL VALUE, UNITS) MAIN 70
C =====MAIN 80
C A1...A18 EXTRA FUNCTION VARIABLES USED IN MATERIAL MAIN 90
C AND COOLANT FUNCTIONS MAIN 100
C =====MAIN 110
C CONR(I,J) INVERSE LOCAL RADIAL-X THERMAL CONDUCTIVITY MAIN 120
C (HR-FT-R/3TU) MAIN 130
C 1/RCOV--(X) (IF RCON--(X)=0.0, CONR(I,J)=1.0E+6) MAIN 140
C CONZ(I,J) INVERSE LOCAL AXIAL-Y-THETA THERMAL MAIN 150
C CONDUCTIVITY (HR-FT-R/3TU) MAIN 160
C 1/ACOV--(X) (IF ACAN--(X)=0.0, CONT(I,J)=1.0E+6) MAIN 170
C CURTI CURRENT TIME (HR) MAIN 180
C =====MAIN 190
C JATI TIME AT WHICH THE TIME DEPENDENT DATA IS MAIN 200
C CALCULATED - MIDDLE OF TIME STEP (HR) MAIN 210
C DELR(I) DISTANCE BETWEEN RADIAL-X POINTS (I) AND MAIN 220
C (I+1) (FT) MAIN 230
C DELZ(J) DISTANCE BETWEEN AXIAL-Y-THETA POINTS (J) MAIN 240
C AND (J+1) (FT) MAIN 250
C JR LOCAL TEMPERATURE (R). USED IN DATA FUNCTIONS. MAIN 260
C DT CURRENT TIME INCREMENT - TIME STEP/2.0 (HR) MAIN 270
C DTIME(M),DISTAN TIME STEP USED IN TIME PERIOD M (HR) MAIN 280
C =====MAIN 290
C FIRSI LOGICAL PARAMETER IN SUBROUTINES MADATA AND MAIN 300
C FLODAT CONTROLLING CERTAIN ACTIONS MAIN 310
C = TRUE, SPECIAL ACTION - ONLY ONCE MAIN 320
C = FALSE, NORMAL ACTION - ALL OTHER TIMES MAIN 330
C FLIM1(N) LOWER LIMIT OF LOW RANGE FLOW RATE FUNCTION, MAIN 340
C FLO--A(X) MAIN 350
C FLIM2(N) UPPER LIMIT OF LOW RANGE FLOW RATE FUNCTION, MAIN 360
C FLO--A(X) MAIN 370
C FLIM3(N) UPPER LIMIT OF MIDDLE RANGE FLOW RATE FUNCTION, MAIN 380
C FLO--B(X) MAIN 390
C FLIM4(N) UPPER LIMIT OF UPPER RANGE FLOW RATE FUNCTION, MAIN 400
C FLO--C(X) MAIN 410
C FLODEP(N) FLOW DEPENDENCE OF COOLANT N MAIN 420
C =0.0, NO DEPENDENCE, USE FLO--A(X) ALWAYS MAIN 430
C =1.0, TIME DEFINES THE FLOW LIMITS (HR) MAIN 440
C =2.0, COOLANT OUTLET TEMPERATURE DEFINES THE MAIN 450

```

C		FLOW LIMITS (F)	MAIN 460
C		=3.0, COOLANT INLET TEMPERATURE DEFINES THE FLOW	MAIN 470
C		LIMITS (F)	MAIN 480
C	FLOW(N)	FLOW RATE OF COOLANT N (LR/HR)	MAIN 490
C		=0, CURRENT COOLANT NOT USED	MAIN 500
C		=1.0E-6, CURRENT COOLANT USED, ZERO FLOW RATE	MAIN 510
C		NOT ALLOWED	MAIN 520
C	FR = FLOW(N)	LOCAL FLOW RATE (LR/HR). USED IN DATA FUNCTIONS.	MAIN 530
C	FT	INDICATOR FOR UNITS OF CURRENT TIME PERIOD	MAIN 540
C		DATA BEING READ	MAIN 550
C		=0.0, SAME UNITS AS PREVIOUS CARD	MAIN 560
C		=1.0, SECONDS	MAIN 570
C		=2.0, MINUTES	MAIN 580
C		=3.0, HOURS	MAIN 590
C	FTIME(M),ENDTI	TIME AT WHICH TIME PERIOD M ENDS (HR)	MAIN 600
C	FTR = RP(I)	RADIAL-X LOCATION OF CURRENT POINT (FT)	MAIN 610
C	FTZ = ZP(J)	AXIAL-Y-THETA LOCATION OF CURRENT POINT (FT)	MAIN 620
C=	=====	=====	MAIN 630
C	GAPX(IG,J)	RADIAL-X GAP INDICATOR	MAIN 640
C		=POSITIVE, LOCAL RADIAL-X GAP WIDTH IN GAPLINE	MAIN 650
C		IG (FT)	MAIN 660
C		=0.0, NO GAP OR COOLANT	MAIN 670
C		=-1.0E-10, COOLANT ON HIGH INDEX SIDE OF GAPLINE	MAIN 680
C		IG	MAIN 690
C		=-2.0E-10, COOLANT ON LOW INDEX SIDE OF GAPLINE	MAIN 700
C		IG	MAIN 710
C		=-3.0E-10, TWO COOLANTS ADJACENT AT GAPLINE IG	MAIN 720
C	GAPZ(JG,I)	AXIAL-Y-THETA GAP INDICATOR	MAIN 730
C		=POSITIVE, LOCAL AXIAL-Y-THETA GAP WIDTH IN	MAIN 740
C		GAPLINE JG (FT)	MAIN 750
C		=0.0, NO GAP OR COOLANT	MAIN 760
C		=-1.0E-10, COOLANT ON HIGH INDEX SIDE OF GAPLINE	MAIN 770
C		JG	MAIN 780
C		=-2.0E-10, COOLANT ON LOW INDEX SIDE OF GAPLINE	MAIN 790
C		JG	MAIN 800
C		=-3.0E-10, TWO COOLANTS ADJACENT AT GAPLINE JG	MAIN 810
C	GAS	LOGICAL VARIABLE IN SUBROUTINE MADATA	MAIN 820
C		=TRUE, EVALUATE GAS CONDUCTIVITY	MAIN 830
C	GK	LOCAL GAS CONDUCTIVITY GCON--(X) (BTU/HR-FT-F)	MAIN 840
C=	=====	=====	MAIN 850
C	HEIGHT	HEIGHT OF SLAB (FT). IF HEIGHT NOT SPECIFIED,	MAIN 860
C		UNIT HEIGHT.	MAIN 870
C	HR	TIME WHEN TIME DEPENDENT DATA CALCULATED -	MAIN 880
C		MIDDLE OF TIME STEP (HR). USED IN DATA FUNCTIONS	MAIN 890
C=	=====	=====	MAIN 900
C	I	RADIAL-X POINT OR GRID LINE INDEX	MAIN 910

C	IA=SELECT	LOCAL HEAT TRANSFER CORRELATION RANGE INDICATOR.	MAIN 920
C		USED IN DATA FUNCTIONS.	MAIN 930
C		=1, EVALUATE HEAT TRANSFER COEFFICIENT FOR THE	MAIN 940
C		LOW REYNOLDS NUMBER RANGE H--A(X)	MAIN 950
C		=2, EVALUATE HEAT TRANSFER COEFFICIENT FOR THE	MAIN 960
C		MIDDLE REYNOLDS NUMBER RANGE H--B(X)	MAIN 970
C		=3, EVALUATE HEAT TRANSFER COEFFICIENT FOR THE	MAIN 980
C		UPPER REYNOLDS NUMBER RANGE H--C(X)	MAIN 990
C	ICOUNT	NUMBER OF RECORDS ON THE EXTRA OUTPUT TAPE	MAIN1000
C	IEKOR(2)	NAME OF SUBROUTINE WHERE ERROR OCCURRED	MAIN1010
C	IF(I)	INTEGER VALUE OF FLOW(N)	MAIN1020
C	IFLOW(N)	INTEGER FLOW DEPENDENCE NUMBER OF COOLANT N	MAIN1030
C		=1, NO DEPENDENCE, USE FLOW--A(X) ALWAYS	MAIN1040
C		=2, TIME DEFINES THE FLOW LIMITS (HR)	MAIN1050
C		=3, COOLANT OUTLET TEMPERATURE DEFINES THE	MAIN1060
C		FLOW LIMITS (F)	MAIN1070
C		=4, COOLANT INLET TEMPERATURE DEFINES THE FLOW	MAIN1080
C		LIMITS (F)	MAIN1090
C	IG	RADIAL-X GAP INDEX	MAIN1100
C	IGH5	THE GAP NUMBER ASSOCIATED WITH THE HIGH RADIAL-X	MAIN1110
C		GRID LINE BOUNDARY OF THE CURRENT BLOCK	MAIN1120
C	IGL5	THE GAP NUMBER ASSOCIATED WITH THE LOW RADIAL-X	MAIN1130
C		GRID LINE BOUNDARY OF THE CURRENT BLOCK	MAIN1140
C	IGR(I)	INDEX OF GAPLINE AT RADIAL-X GRID LINE (I)	MAIN1150
C	IH(L)	INDEX OF THE HIGH RADIAL-X POINT OF BLOCK L	MAIN1160
C	IHS = IH(L)	INDEX OF THE HIGH RADIAL-X POINT OF CURRENT	MAIN1170
C		BLOCK	MAIN1180
C	IL(L)	INDEX OF THE LOW RADIAL-X POINT OF BLOCK L	MAIN1190
C	ILS = IL(L)	INDEX OF THE LOW RADIAL-X POINT OF CURRENT	MAIN1200
C		BLOCK	MAIN1210
C	IM	NUMBER OF RADIAL-Y GRID LINES	MAIN1220
C	IM1	IM-1	MAIN1230
C	IMAX	NUMBER OF RADIAL-Y POINTS	MAIN1240
C	IP = IFLOW(N)	LOCAL FLOW DEPENDENCE. USED IN DATA FUNCTIONS.	MAIN1250
C	IPATH(N)	FLOW DIRECTION FOR COLLANT N	MAIN1260
C		=1, RADIAL-X DIRECTION	MAIN1270
C		=2, AXIAL-Y-THETA DIRECTION	MAIN1280
C		=3, NORMAL TO SLAB	MAIN1290
C		=+, FLOW IN DIRECTION OF INCREASING INDEX	MAIN1300
C		=-, FLOW IN DIRECTION OF DECREASING INDEX	MAIN1310
C	ISHAPE	GEOMETRY TYPE INDICATOR	MAIN1320
C		=0, CYLINDRICAL GEOMETRY	MAIN1330
C		=1, RECTANGULAR GEOMETRY	MAIN1340
C		=2, CIRCULAR GEOMETRY	MAIN1350
C	IT = ITIN(N)	LOCAL INLET TEMPERATURE DEPENDENCE	MAIN1360
C	ITAPE(M)	NUMBER OF TIMESTEPS BETWEEN TEMPERATURE	MAIN1370

C		PRINTS IN PERIOD M	MAIN1380
C	ITER(M)	NUMBER OF ITERATIONS BETWEEN PRINTOUTS IN	MAIN1390
C		TIME PERIOD M	MAIN1400
C	ITI(N)	INTEGER VALUE OF T _i (N)	MAIN1410
C	ITIN(N)	INTEGER INLET TEMPERATURE DEPENDENCE OF COOLANT	MAIN1420
C		N	MAIN1430
C		=1, NO DEPENDENCE, USE TIN--A(X) ALWAYS	MAIN1440
C		=2, TIME DEFINES THE INLET TEMPERATURE LIMITS	MAIN1450
C		(HR)	MAIN1460
C		=3, FLOW DEFINES THE INLET TEMPERATURE LIMITS	MAIN1470
C		(LB/HR)	MAIN1480
C		=4, OUTLET TEMPERATURE DEFINES THE INLET	MAIN1490
C		TEMPERATURE LIMITS (F)	MAIN1500
C	ITO(N)	INTEGER VALUE OF T _o (N)	MAIN1510
C	IUNIT	INTEGER VALUE OF F _T , CODE NUMBER INDICATING THE	MAIN1520
C		UNITS OF THE CURRENT TIME PERIOD DATA BEING READ	MAIN1530
C=	=====	=====	MAIN1540
C	J	AXIAL-Y-THETA POINT OR GRID LINE INDEX	MAIN1550
C	JG	AXIAL-Y-THETA GAP INDEX	MAIN1560
C	JGZ(J)	INDEX OF GAPLINE AT AXIAL-Y-THETA GRID LINE (J)	MAIN1570
C	JM1	JM-1	MAIN1580
C	JGHS	THE GAP NUMBER ASSOCIATED WITH THE HIGH AXIAL-Y-	MAIN1590
C		THETA GRID LINE BOUNDARY OF THE CURRENT BLOCK	MAIN1600
C	JGLS	THE GAP NUMBER ASSOCIATED WITH THE LOW AXIAL-Y-	MAIN1610
C		THETA GRID LINE BOUNDARY OF THE CURRENT BLOCK	MAIN1620
C	JH(L)	INDEX OF THE HIGH AXIAL-Y-THETA POINT OF BLOCK L	MAIN1630
C	JHS = JH(L)	INDEX OF THE HIGH AXIAL-Y-THETA POINT OF	MAIN1640
C		CURRENT BLOCK	MAIN1650
C	JL(L)	INDEX OF THE LOW AXIAL-Y-THETA POINT OF BLOCK L	MAIN1660
C	JLS = JL(L)	INDEX OF THE LOW AXIAL-Y-THETA POINT OF	MAIN1670
C		CURRENT BLOCK	MAIN1680
C	JM	NUMBER OF AXIAL-Y-THETA GRID LINES	MAIN1690
C	JMAX	NUMBER OF AXIAL-Y-THETA POINTS	MAIN1700
C=	=====	=====	MAIN1710
C	KR(1,J)	EFFECTIVE RADIAL-Y CONDUCTIVITY BETWEEN POINTS	MAIN1720
C		(I,J) AND (I+1,J) (BTU/HR-FT-F)	MAIN1730
C	KZ(1,J)	EFFECTIVE AXIAL-Y-THETA CONDUCTIVITY BETWEEN	MAIN1740
C		POINTS (I,J) AND (I,J+1) (BTU/HR-FT-F)	MAIN1750
C=	=====	=====	MAIN1760
C	LMAX	NUMBER OF BLOCKS	MAIN1770
C=	=====	=====	MAIN1780
C	M = IPATH(N)	LOCAL FLOW DIRECTION	MAIN1790
C	MATRG(IG,J)	1. NUMBER OF GAS IN GAP ALONG RADIAL-X GAPLINE	MAIN1800
C		IG AT AXIAL-Y-THETA LEVEL J OR,	MAIN1810
C		2. INDICATOR DENOTING STATUS OF RADIATION ON	MAIN1820
C		RADIAL-X COOLANT BOUNDARY(GAPLINE) IG AT AXIAL-	MAIN1830

C		-Y-THETA LEVEL J	MAIN1840
C		=100, NO RADIATION SPECIFIED	MAIN1850
C		=200, RADIATION SPECIFIED AND INCLUDED	MAIN1860
C		=300, RADIATION SPECIFIED BUT TEMPORARILY	MAIN1870
C		EXCLUDED IN ORDER TO COMPLETE AN	MAIN1880
C		ITERATION	MAIN1890
C	MATZG(JG,I)	1. NUMBER OF GAS IN GAP ALONG AXIAL-Y-THETA	MAIN1900
C		GAPLINE JG AT RADIAL-X LEVEL I OR,	MAIN1910
C		2. INDICATOR DENOTING STATUS OF RADIATION ON	MAIN1920
C		AXIAL-Y-THETA COOLANT BOUNDARY(GAPLINE) JG AT	MAIN1930
C		RADIAL-X LEVEL I	MAIN1940
C		=100, NO RADIATION SPECIFIED	MAIN1950
C		=200, RADIATION SPECIFIED AND INCLUDED	MAIN1960
C		=300, RADIATION SPECIFIED BUT TEMPORARILY	MAIN1970
C		EXCLUDED IN ORDER TO COMPLETE AN	MAIN1980
C		ITERATION	MAIN1990
C	MAXFLD,NQ	MAXIMUM NUMBER OF COOLANTS ALLOWED	MAIN2000
C	MAXMAT,15	MAXIMUM NUMBER OF MATERIALS OR COOLANT ALLOWED	MAIN2010
C	MAXNB,LQ	MAXIMUM NUMBER OF BLOCKS ALLOWED	MAIN2020
C	MAXRG,IQ	MAXIMUM NUMBER OF RADIAL-X GAP LINES	MAIN2030
C	MAXRP,IQ	MAXIMUM NUMBER OF RADIAL-X POINTS ALLOWED	MAIN2040
C	MAXZG,JG	MAXIMUM NUMBER OF AXIAL-Y-THETA GAP LINES	MAIN2050
C	MAXZP,JG	MAXIMUM NUMBER OF AXIAL-Y-THETA POINTS ALLOWED	MAIN2060
C	MB(L)	MATERIAL NUMBER OF BLOCK L	MAIN2070
C		IF COOLANT, COOLANT NUMBER STORED AS NEGATIVE	MAIN2080
C		MATERIAL NUMBER	MAIN2090
C	MGR(L)	MATERIAL NUMBER OF THE RADIAL-X GAP OF BLOCK L	MAIN2100
C	MGZ(L)	MATERIAL NUMBER OF THE AXIAL-Y-THETA GAP OF	MAIN2110
C		BLOCK L	MAIN2120
C	MMAX(20)	MAXIMUM NUMBER OF TIME PERIODS ALLOWED	MAIN2130
C	MJ	MAXIMUM OF IQ,JG	MAIN2140
C	MT(1,J)	TABLE USED TO TEST COMPLETENESS OF THE BLOCK	MAIN2150
C		ASSIGNMENTS	MAIN2160
C	MX	MAXFLD + MAXMAT	MAIN2170
C=	=====	=====	MAIN2180
C	V	COOLANT INDEX	MAIN2190
C	VI	MATERIAL INDEX	MAIN2200
C	VITER	THE CURRENT ITERATION NUMBER. ONE (RADIAL-X)	MAIN2210
C		AND (AXIAL-Y-THETA) PASS ARE COUNTED AS ONE	MAIN2220
C		ITERATION.	MAIN2230
C	VRG	NUMBER OF RADIAL-Y GAPLINES	MAIN2240
C	NUMBER(60)	LIST OF INTEGERS FROM 1 TO 60 USED IN PRINTOUT	MAIN2250
C	VZG	NUMBER OF AXIAL-Y-THETA GAPLINES	MAIN2260
C=	=====	=====	MAIN2270
C	PNAME(12)	ALPHANUMERIC DATA USED IN PRINTOUT HEADINGS	MAIN2280
C=	=====	=====	MAIN2290

C	OUTTAP,15	TAPE ON WHICH OUTPUT DATA IS SAVED	MATN2300
C=	=====	=====	MATN2310
C	RATIOB(IG,J)	GEOMETRICAL CONSTANT USED TO CALCULATE THE	MATN2320
C		EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MATN2330
C	RATIOC(IG)	GEOMETRICAL CONSTANT USED TO CALCULATE THE	MATN2340
C		EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MATN2350
C	RATIOH(IG)	GEOMETRICAL CONSTANT USED TO CALCULATE THE	MATN2360
C		EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MATN2370
C	RATIOK(IG)	GEOMETRICAL CONSTANT USED TO CALCULATE THE	MATN2380
C		EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MATN2390
C	RUBH(IG,J)	LOCAL TEMPERATURE OF THE RADIAL-X BOUNDARY OF	MATN2400
C		A BLOCK BOUNDED ON ITS HIGH I INDEX SIDE BY	MATN2410
C		GAPLINE IG (R)	MATN2420
C	RBBIL(IG,J)	LOCAL TEMPERATURE OF THE RADIAL-X BOUNDARY OF	MATN2430
C		A BLOCK BOUNDED ON ITS HIGH I INDEX SIDE BY	MATN2440
C		GAPLINE IG (R)	MATN2450
C	RBL(L)	LOW RADIAL-X BOUNDARY OF BLOCK L (FT)	MATN2460
C	RBH(L)	HIGH RADIAL-X BOUNDARY OF BLOCK L (FT)	MATN2470
C	RCP(I,J)	SPECIFIC HEAT OF VOLUME (I,J) (BTU/R)	MATN2480
C	RCPL(N,I)	SPECIFIC HEAT OF COOLANT N, LEVEL I IN FLOW	MATN2490
C		DIRECTION (BTU/LB-HR)	MATN2500
C	RDG(L)	RADIAL-X GAP THICKNESS (FT)	MATN2510
C	RE	LOCAL REYNOLDS NUMBER. USED IN DATA FUNCTIONS.	MATN2520
C	REMH(IG,J)	INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL-X	MATN2530
C		BOUNDARY OF A BLOCK BOUNDED ON ITS HIGH I INDEX	MATN2540
C		SIDE BY GAPLINE IG	MATN2550
C		1/EMRH--(X), IF EMRH--(X)=0.0, REMH(IG,J)=1.0E+6	MATN2560
C	REML(IG,J)	INVERSE OF THE LOCAL EMISSIVITY OF THE RADIAL-X	MATN2570
C		BOUNDARY OF A BLOCK BOUNDED ON ITS LOW I INDEX	MATN2580
C		SIDE BY GAPLINE IG	MATN2590
C		1/EMRL--(X), IF EMRL--(X)=0.0, REML(IG,J)=1.0E+6	MATN2600
C	RL(1)	RADIAL-X GRID LINE (FT)	MATN2610
C		IF RL(1)=0.0, THEN RL(1) SET EQUAL TO 0.0833E-5	MATN2620
C	RLIM1(N)	REYNOLDS NUMBER LIMITS SPECIFY THE HEAT TRANSFER	MATN2630
C		CORRELATION	MATN2640
C		LOWER LIMIT OF LOW RANGE HEAT TRANSFER	MATN2650
C		CORRELATION, H--A(Y)	MATN2660
C	RLIM2(N)	UPPER LIMIT OF LOW RANGE HEAT TRANSFER	MATN2670
C		CORRELATION, H--A(Y)	MATN2680
C	RLIM3(N)	UPPER LIMIT OF MIDDLE RANGE HEAT TRANSFER	MATN2690
C		CORRELATION, H--B(Y)	MATN2700
C	RLIM4(N)	UPPER LIMIT OF UPPER RANGE HEAT TRANSFER	MATN2710
C		CORRELATION, H--C(Y)	MATN2720
C	RLN(I)	GEOMETRICAL CONSTANT USED TO CALCULATE THE	MATN2730
C		EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARY	MATN2740
C	RP(I)	RADIAL-X LOCATION OF POINT I (FT)	MATN2750

C	RR(1,J)	RADIAL-X GEOMETRY FACTOR (HEAT FLOW AREA/	MAIN2760
C		RADIAL-X DISTANCE, BETWEEN POINTS (I,J) AND	MAIN2770
C		(I+1,J) (FT)	MAIN2780
C	RZ(1,J)	AXIAL-Y-THETA GEOMETRY FACTOR (HEAT FLOW AREA/	MAIN2790
C		AXIAL-Y-THETA DISTANCE) BETWEEN POINTS (I,J)	MAIN2800
C		AND (I,J+1) (FT)	MAIN2810
C=	=====	=====	MAIN2820
C	SCALE	CONVERSION FACTOR	MAIN2830
C		=12.0, CYLINDRICAL AND RECTANGULAR GEOMETRY	MAIN2840
C		=12.0, RECTANGULAR GEOMETRY	MAIN2850
C		=67.2957795, CYLINDRICAL AND CIRCULAR GEOMETRY	MAIN2860
C	ST	LOCAL SURFACE TEMPERATURE (R). USED IN DATA	MAIN2870
C		FUNCTIONS.	MAIN2880
C	SW(20)	LOGICAL PARAMETER INDICATING OPTIONS REQUESTED	MAIN2890
C=	=====	=====	MAIN2900
C	T(I,J)	LOCAL TEMPERATURE OF POINT (I,J) (R)	MAIN2910
C	TB	LOCAL WALL TEMPERATURE (R)	MAIN2920
C	TH	LOCAL WALL TEMPERATURE (R)	MAIN2930
C	TI(N)	COOLANT INLET TEMPERATURE (R)	MAIN2940
C	TIN = TI(N)	LOCAL COOLANT INLET TEMPERATURE (R). USED	MAIN2950
C		IN DATA FUNCTIONS.	MAIN2960
C	TINDEP(N)	INLET TEMPERATURE DEPENDENCE OF COOLANT N	MAIN2970
C		=0.0, NO DEPENDENCE, USE TIN=A(X) ALWAYS	MAIN2980
C		=1.0, TIME DEFINES THE INLET TEMPERATURE LIMITS	MAIN2990
C		(HR)	MAIN3000
C		=2.0, FLOW DEFINES THE INLET TEMPERATURE LIMITS	MAIN3010
C		(LB/HR)	MAIN3020
C		=3.0, OUTLET TEMPERATURE DEFINES THE INLET	MAIN3030
C		TEMPERATURE LIMITS (F)	MAIN3040
C	TLIM1(N)	LOWER LIMIT OF LOW RANGE INLET TEMPERATURE	MAIN3050
C		FUNCTION, TIN=A(X),	MAIN3060
C	TLIM2(N)	UPPER LIMIT OF LOW RANGE INLET TEMPERATURE	MAIN3070
C		FUNCTION, TIN=A(X),	MAIN3080
C	TLIM3(N)	UPPER LIMIT OF MIDDLE RANGE INLET TEMPERATURE	MAIN3090
C		FUNCTION, TIN=B(X),	MAIN3100
C	TLIM4(N)	UPPER LIMIT OF UPPER RANGE INLET TEMPERATURE	MAIN3110
C		FUNCTION, TIN=C(X),	MAIN3120
C	TO(N)	COOLANT OUTLET TEMPERATURE (R)	MAIN3130
C	TOU1 = TO(N)	LOCAL COOLANT OUTLET TEMPERATURE (R). USED	MAIN3140
C		IN DATA FUNCTIONS.	MAIN3150
C	TT(1,J)	INTEGER VALUE OF T(I,J) (F)	MAIN3160
C		CONVERSION FACTOR RANKINE TO FAHRENHEIT = 459.5	MAIN3170
C=	=====	=====	MAIN3180
C	W(I,J)	HEAT GENERATION OF VOLUME (I,J) (BTU/HR)	MAIN3190
C=	=====	=====	MAIN3200
C	V(I,J)	NODE VOLUME FOR POINT (I,J) (FT**3)	MAIN3210

```

C          IF HEIGHT NOT SPECIFIED, THEN UNIT HEIGHT.          MAIN3220
C= =====MAIN3230
C      ZA(12)          FIRST TITLE CARD, USED AS PAGE HEADING          MAIN3240
C      ZAT10B(JG,I)    GEOMETRICAL CONSTANT USED TO CALCULATE THE          MAIN3250
C                      EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3260
C      ZAT10H(JG)      GEOMETRICAL CONSTANT USED TO CALCULATE THE          MAIN3270
C                      EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3280
C      ZAT10K(JG)      GEOMETRICAL CONSTANT USED TO CALCULATE THE          MAIN3290
C                      EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3300
C      ZBB1H(JG,I)     LOCAL TEMPERATURE OF THE AXIAL-Y-THETA BOUNDARYMAIN3310
C                      OF A BLOCK BOUNDED ON ITS HIGH J INDEX SIDE          MAIN3320
C                      BY GAPLINE JG (R)          MAIN3330
C      ZBB1L(JG,I)     LOCAL TEMPERATURE OF THE AXIAL-Y-THETA BOUNDARYMAIN3340
C                      OF A BLOCK BOUNDED ON ITS LOW J INDEX SIDE          MAIN3350
C                      BY GAPLINE JG (R)          MAIN3360
C      ZBH(L)          HIGH AXIAL-Y-THETA BOUNDARY OF BLOCK L (FT)          MAIN3370
C      ZBL(L)          LOW AXIAL-Y-THETA BOUNDARY OF BLOCK L (FT)          MAIN3380
C      ZDG(L)          AXIAL-Y-THETA GAP THICKNESS (FT)          MAIN3390
C      ZEMH(JG,I)      INVERSE OF THE LOCAL EMISSIVITY OF THE AXIAL-Y-MAIN3400
C                      THETA BOUNDARY OF A BLOCK BOUNDED ON ITS LOW J          MAIN3410
C                      INDEX SIDE BY GAPLINE JG          MAIN3420
C                      1/EMAH--(X), IF EMAL--(X)=0.0, ZEMH(JG,I)=1.0E+6MAIN3430
C      ZEML(JG,I)      INVERSE OF THE LOCAL EMISSIVITY OF THE AXIAL-Y-MAIN3440
C                      THETA BOUNDARY OF A BLOCK BOUNDED ON ITS LOW J          MAIN3450
C                      INDEX SIDE BY GAPLINE JG          MAIN3460
C                      1/EMAL--(X), IF EMAL--(X)=0.0, ZEML(JG,I)=1.0E+6MAIN3470
C      ZL(J)          AXIAL-Y-THETA GRID LINE (FT)          MAIN3480
C      ZLN(J)          GEOMETRICAL CONSTANT USED TO CALCULATE THE          MAIN3490
C                      EFFECTIVE THERMAL CONDUCTIVITY ACROSS A BOUNDARYMAIN3500
C      ZP(J)          AXIAL-Y-THETA LOCATION OF POINT J (FT)          MAIN3510
C= =====MAIN3520
C= =====MAIN3530
C      READ AND PROCESS THE INPUT DATA OF THE NEXT PROBLEM          MAIN3540
100 CALL MP1          MAIN3550
C      HAS ANY ERROR BEEN FOUND? -'YES,NO'-          MAIN3560
C      IF (IERROR(1).NE.0) GO TO 100          MAIN3570
C      CALCULATE AND PRINT THE TEMPERATURE DISTRIBUTIONS          MAIN3580
C      CALL MP2          MAIN3590
C      GO TO 100          MAIN3600
C= =====MAIN3610
C      END          MAIN3620

```

	SUBROUTINE MP1	MP1	10
	INCLUDE COMJIM	MP1	20
C=	=====	MP1	30
C0	READ AND PROCESS THE INPUT DATA OF ONE PROBLEM	MP1	40
C=	=====	MP1	50
C=	=====	MP1	60
C	ASSIGN THE FIXED CONSTANTS	MP1	70
	OUTIAP=15	MP1	80
	MAXRP=IQ	MP1	90
	MAXZP=JQ	MP1	100
	MAXRG=IGQ	MP1	110
	MAXZG=JGQ	MP1	120
	MAXMAT=15	MP1	130
	MAXFLO=NG	MP1	140
	MAXV3=LQ	MP1	150
	MMAX=20	MP1	160
	WRITE (0,100) MAXRP,MAXZP,MAXRG,MAXZG,MAXNB	MP1	170
100	FORMAT(1H1,9X,19HT A C 2 D C O D E,5X,16HSTANDARD VERSION/30	MP1	180
	1X,5H1Q = ,I2,4X,5HJQ = ,I2,4X,6HIQ - ,I2,4X,6HJGQ = ,I2,4X,5HLQ	MP1	190
	2 13////)	MP1	200
C=	=====	MP1	210
	IERROR(1)=0	MP1	220
	IERROR(2)=0	MP1	230
	ICOUNT=0	MP1	240
	HEIGHT=1.0	MP1	250
	DO 130 I=1,25	MP1	260
	SW(I)=.FALSE.	MP1	270
130	CONTINUE	MP1	280
C=	=====	MP1	290
C	READ AND PRINT THE TITLE AND GEOMETRICAL DATA	MP1	300
	CALL INPUT	MP1	310
C	CHECK THE GEOMETRICAL DATA	MP1	320
	CALL CHECK	MP1	330
C	ASSIGN THE POINTS, BLOCKS AND GAPS	MP1	340
	CALL POINTS	MP1	350
C	CALCULATE THE CONSTANTS USED TO CORRECT FOR GAPS AND COOLANTS	MP1	360
	CALL CONSTA	MP1	370
C	CALCULATE THE GEOMETRICAL PART OF THE RESISTANCES BETWEEN POINTS	MP1	380
	CALL GEOMET	MP1	390
C	PRINT THE LOCATION OF POINTS AND BOUNDARIES	MP1	400
	CALL PRINT	MP1	410
C	READ THE INITIAL TEMPERATURE DISTRIBUTION	MP1	420
	CALL INITEM	MP1	430
C	READ AND PRINT THE COOLANT DATA	MP1	440
	CALL FLOWCA	MP1	450

C	READ AND PRINT THE TIMESTEP SEQUENCE	MP1	460
	CALL TIME	MP1	470
C	READ AND PRINT THE GROUP OF SPECIAL FUNCTION PARAMETERS	MP1	480
	CALL EXTRA	MP1	490
C	READ THE PREVIOUS TEMPERATURE DISTRIBUTION, IF PRESENT	MP1	500
	CALL PRETEM	MP1	510
	CALL UTCARD	MP1	520
C=	=====	MP1	530
C	STOP EXECUTION IF ANY ERRORS HAVE BEEN FOUND	MP1	540
	IF (IERROR(1).NE.0) STOP	MP1	550
C=	=====	MP1	560
	DO 140 IG=1,MAXRG	MP1	570
	DO 140 J=1,MAXZP	MP1	580
	REML(IG,J)=0.0	MP1	590
	REMH(IG,J)=0.0	MP1	600
140	CONTINUE	MP1	610
	DO 150 JG=1,MAXZG	MP1	620
	DO 150 I=1,MAXRP	MP1	630
	ZEML(JG,I)=0.0	MP1	640
	ZEMH(JG,I)=0.0	MP1	650
150	CONTINUE	MP1	660
	NC=0	MP1	670
	DO 293 I=1,MAXFLO	MP1	680
	IF (NFLO(I).EQ. 0) GO TO 294	MP1	690
	NC=NC+1	MP1	700
293	CONTINUE	MP1	710
294	AC=NC/2.0	MP1	720
	N=NC/2	MP1	730
	IF ((AC-N).GT.0.3) N=N+1	MP1	740
	NC=33-N	MP1	750
	NC=NC/2	MP1	760
C=	=====	MP1	770
C	SHOULD THE OUTPUT BE PUT ON A TAPE? _'NO,YES'=_	MP1	780
	IF (.NOT.SW(3)) RETURN	MP1	790
C	PUT THE INITIAL RECORD (TITLE,GRID INFORMATION) ON THE TAPE	MP1	800
	ILEN=1	MP1	810
	WRITE (OUTTAP) ISHAPE,ILEN,IMAX,JMAX,LMAX,MAXFLO,NRG,N7G,(ZA(I),I=MP1	820	
	11,12),(RL(I),I=1,IMAX),(ZL(J),J=1,JMAX),(IGR(I),I=1,IMAX),(JGZ(J),MP1	830	
	2J=1,JMAX),(IL(K),IH(K),JL(K),JH(K),Ma(K),K=1,LMAX)	MP1	840
C=	=====	MP1	850
	RETURN	MP1	860
C*	*****	MP1	870
C*	*****	MP1	880
C*	*****	MP1	890
	SUBROUTINE UTCARD	MP1	900
	DIMENSION FORMT (8),FORM (8),FB (2)	MP1	910

DATA FORMT	/6H(10A,, 6*6H 12X,, 6HA6,A2)/	MP1	920
DATA FB	/6HF12.5,, 6HE12.5,/	MP1	930
WRITE (6,90)(ZA(I),I=1,12)		MP1	940
90 FORMAT('1H1,30X,12A6,///,40X,21HPROBLEM INPUT CARDS,///,10X,		80HMP1	950
1 0 1 1 2 2 3 3 4 4 5 5 6		6 MP1	960
2 / 7 8,/,10X,		80HMP1	970
3 5 0 5 0 5 0 5 0 5 0 5 0		5 MP1	980
4 0 5 0,/,10X,		80HMP1	990
5XXXXXXAXXX		MP1	1000
6XXXXXXXXXXXXX,///)		MP1	1010
IC=0		MP1	1020
100 IC=IC+1		MP1	1030
IF (IC.GT.NC) RETURN		MP1	1040
ITYP=CARJ(IC)+0.1		MP1	1050
IF (ITYP.EQ.0) GO TO 150		MP1	1060
IF (ITYP.EQ.7) GO TO 160		MP1	1070
DO 120 I=1,8		MP1	1080
FORM(I)=FORM(I)		MP1	1090
120 CONTINUE		MP1	1100
DO 130 J=1,ITYP		MP1	1110
FORM(I+1)=FB(1)		MP1	1120
J=IC+1		MP1	1130
IF (CARD(J).NE.0.0) GO TO 125		MP1	1140
CARD(J)=0.0		MP1	1150
GO TO 130		MP1	1160
125 IF (CARD(J) .LT. 0.0001 .OR. CARD(J) .GT. 9999.9999)FORM(I+1)=FB(2)		MP1	1170
130 CONTINUE		MP1	1180
140 I1=IC+1		MP1	1190
I2=IC+ITYP+2		MP1	1200
WRITE (6,FORM) (CARD(I),I=I1,I2)		MP1	1210
GO TO 160		MP1	1220
150 WRITE (6,155)		MP1	1230
155 FORM(I(62X,5HBLANK))		MP1	1240
GO TO 100		MP1	1250
160 I1=IC+1		MP1	1260
I2=IC+14		MP1	1270
WRITE (6,170) (CARD(I),I=I1,I2)		MP1	1280
170 FORMAT (10X,13A6,A2)		MP1	1290
180 IC=12		MP1	1300
GO TO 100		MP1	1310
RETURN		MP1	1320
C* *****		MP1	1330
C* *****		MP1	1340
END		MP1	1350

```

SUBROUTINE INPUT                                     INPU 10
INCLUDE COMDIM                                     INPU 20
=====INPU 30
C READ THE GEOMETRY DATA.                         INPU 40
C=====INPU 50
C ERROR STOPS=                                     INPU 60
C   INPUT1 THE GEOMETRY TYPE DESIRED HAS BEEN MISPELLED. INPU 70
C   INPUT2 NORMAL FLOW NOT ALLOWED IN CYLINDRICAL GEOMETRY INPU 80
C   INPUT3 THE LOW RADIAL-X BLOCK BOUNDARY OF SOME BLOCK DOES INPU 90
C           NOT COINCIDE WITH A RADIAL-X GRID LINE INPU 100
C   INPUT4 THE HIGH RADIAL-X BLOCK BOUNDARY OF SOME BLOCK DOES INPU 110
C           NOT COINCIDE WITH A RADIAL-X GRID LINE INPU 120
C   INPUT5 THE LOW AXIAL-Y-THETA BLOCK BOUNDARY OF SOME BLOCK INPU 130
C           DOES NOT COINCIDE WITH A AXIAL-Y-THETA GRID LINE INPU 140
C   INPUT6 THE HIGH AXIAL-Y-THETA BLOCK BOUNDARY OF SOME BLOCK INPU 150
C           DOES NOT COINCIDE WITH A AXIAL-Y-THETA GRID LINE INPU 160
C=====INPU 170
C=====INPU 180
C INTEGER      BLANKS      ,OPT  ( 21),OPTION( 14) INPU 190
C DIMENSION    TYPE  ( 10),BTABLE ( 10,JQ) INPU 200
C EQUIVALENCE  (BTABLE ,RCP  ) INPU 210
C DATA (OPT(I), I = 1,21) INPU 220
1      /4HJUMP, 5HPUNCH, 4HYPE, 6HRESIST, 6HALL HE, INPU 230
2      6HALL SU, 6HALL CO, 4HTIMEST, 6HALL DE, 6HSTEADY, INPU 240
3      6HHEAT F, 6HSURFAC, 4HCONJUC, 6HRECTAN, 6HCYLIND, INPU 250
4      6HCIRCUL, 6HRADIAL, 4HAXIAL, 5HTHETA, 6HDECIMA, INPU 260
5      6HINVERS/ INPU 270
C DATA BLANKS /6H / INPU 280
C DATA (TYPE(I),I=1,10) INPU 290
1      /6HRADIAL, 5HAXIAL, 4HTHETA, 4H X, 4H Y, INPU 300
2      6H(INCHE, 2HS), 6H(DFGRE, 3HES), 6HNORMAL/ INPU 310
C DATA BLANK /6H / INPU 320
C DATA STAR /5H* / INPU 330
C DATA GAP /5H# / INPU 340
C DATA DOT /5H. / INPU 350
C=====INPU 360
C PNAME(3)=TYPE(10) INPU 370
C PNAME(4)=TYPE(6) INPU 380
C PNAME(5)=TYPE(7) INPU 390
C PNAME(6)=TYPE(6) INPU 400
C PNAME(7)=TYPE(7) INPU 410
C NC=0 INPU 420
C=====INPU 430
C READ AND PRINT THE TITLE INPU 440
C CALL TITLE INPU 450

```

```

C= =====INPU 460
C READ THE NEXT OPTION INPU 470
100 READ (5,110) (OPTION(I),I=1,14) INPU 480
110 FORMAT (13A6,A2) INPU 490
WRITE (6,120) (OPTION(I),I=1,12) INPU 500
120 FORMAT (1H0,19X,12A6) INPU 510
C HAVE ALL OPTIONS BEEN READ? -'YES,NO,- INPU 520
IF (OPTION(2).EQ.BLANKS) GO TO 150 INPU 530
CALL ACARD(OPTION) INPU 540
C/ ASSIGN THE OPTION -'OK,-RROR'- INPU 550
DO 130 I=1,21 INPU 560
IF (OPTION(2).NE.OPT(I)) GO TO 130 INPU 570
SW(1)=.TRUE. INPU 580
GO TO 100 INPU 590
130 CONTINUE INPU 600
CE *****INPU 610
C OPTION COULD NOT BE FOUND, PRINT A WARNING INPU 620
WRITE (6,140) OPTION(2) INPU 630
140 FORMAT (8H1OPTION 'A6,1H COULD NOT BE READ) INPU 640
GO TO 100 INPU 650
C= =====INPU 660
C SPECIFY THE PRINT FORMAT INPU 670
150 NC=NC+1 INPU 680
CARD(NC)=0.0 INPU 690
C IS IT A RECTANGULAR GEOMETRY? -'YES,NO'- INPU 700
IF (SW(14)) GO TO 170 INPU 710
C IS IT A CYLINDRICAL GEOMETRY? -'YES,NO'- INPU 720
IF (SW(15)) GO TO 160 INPU 730
C IS IT A CIRCULAR GEOMETRY? -'YES,NO'- INPU 740
IF (SW(16)) GO TO 180 INPU 750
C THE TYPE OF GEOMETRY IS NOT SPECIFIED CORRECTLY INPU 760
CALL ERROR (6HINPUT1) INPU 770
C= =====INPU 780
C SET THE PROBLEM UP FOR A CYLINDRICAL GEOMETRY INPU 790
160 SCALE=12. INPU 800
ISHAPE=0 INPU 810
PNAME(1)=TYPE(1) INPU 820
PNAME(2)=TYPE(2) INPU 830
GO TO 190 INPU 840
C= =====INPU 850
C SET THE PROBLEM UP FOR A RECTANGULAR GEOMETRY INPU 860
170 SCALE=12.0 INPU 870
ISHAPE=1 INPU 880
PNAME(1)=TYPE(4) INPU 890
PNAME(2)=TYPE(5) INPU 900
GO TO 190 INPU 910

```

```

C= =====INPU 920
C   SET UP THE PROBLEM FOR A CIRCULAR GEOMETRY                               INPU 930
180 SCALE=57.2957795                                                         INPU 940
    ISHAPE=2                                                                  INPU 950
    PNAME(1)=TYPE(1)                                                         INPU 960
    PNAME(2)=TYPE(3)                                                         INPU 970
    PNAME(4)=TYPE(8)                                                         INPU 980
    PNAME(5)=TYPE(9)                                                         INPU 990
C   =====INPU1000
CB  READ AND PROCESS THE GRID DESCRIPTIONS                                  INPU1010
190 IM=MAXRP-1                                                                INPU1020
C   READ THE INPUT DESCRIBING THE RADIAL-X GRID LINES                      INPU1030
    CALL GRID (RL,IM)                                                         INPU1040
    JM=MAXZP-1                                                                INPU1050
C   READ THE INPUT DESCRIBING THE AXIAL-Y-THETA GRID LINES                INPU1060
    CALL GRID (ZL,JM)                                                         INPU1070
C= =====INPU1080
C   ASSIGN THE GRID LIMITS                                                  INPU1090
    IMAX=IM+1                                                                INPU1100
    JMAX=JM+1                                                                INPU1110
    IM1=IM-1                                                                INPU1120
    JM1=JM-1                                                                INPU1130
C= =====INPU1140
C   CHANGE A ZERO INNER RADIUS TO A SMALL BUT FINITE ONE                  INPU1150
    IF (RL(1).LE.0.0) RL(1)=1.0E-5                                         INPU1160
C   CHANGE THE RADIAL-X GRID LINE DATA TO THE CORRECT UNITS              INPU1170
    DO 200 I=1,IM                                                            INPU1180
        RL(I)=RL(I)/12.0                                                    INPU1190
200 CONTINUE                                                                INPU1200
C= =====INPU1210
C   CHANGE THE AXIAL-Y-THETA GRID LINE DATA TO THE CORRECT UNITS        INPU1220
    DO 210 J=1,JM                                                            INPU1230
        ZL(J)=ZL(J)/SCALE                                                  INPU1240
210 CONTINUE                                                                INPU1250
C= =====INPU1260
C   PRINT THE HEADING OF THE BLOCK PRINT                                  INPU1270
    WRITE (6,220) (ZA(I),I=1,12),PNAME(1),PNAME(1),PNAME(2),PNAME(2),PINPU1280
    1NAME(1),PNAME(2),PNAME(4),PNAME(5),PNAME(4),PNAME(5),PNAME(4),PNAMINPU1290
    2E(5)                                                                    INPU1300
220 FORMAT (1H1,30X,12A6,///,50X,18HBLOCK, DESCRIPTION,///,3X,5HBLOCK,6INPU1310
1X,50H. . . . . BOUNDARIES . . . . .,7XINPU1320
2,36H. . . . . GAPS . . . . . /2X,6HNUMBER/13X,4HLOW ,A6INPU1330
3,3X,5HHIGH ,A6,4X,4HLOW ,A6,4X,5HHIGH ,A6,10H MATERIAL,5X,A6,10H INPU1340
4 MATERIAL,4X,A6,10H MATERIAL/14X,8H,INCHES),5X,8H(INCHES),7X,2A6,INPU1350
52X,2A6,12X,8H(INCHES),12X,2A6//)                                         INPU1360
C= =====INPU1370

```


CB	READ AND PRINT THE BLOCK DESCRIPTIONS	INPU1380
	K=0	INPU1390
C	COUNT THE BLOCKS	INPU1400
230	K=K+1	INPU1410
C	READ THE MATERIAL NUMBER	INPU1420
	READ (5,350) X,Y,Z,FLAG1,FLAG2	INPU1430
C	HAVE ALL THE BLOCK CARDS BEEN READ? -'YES,NO'-	INPU1440
	IF (X.EQ.0.0) GO TO 250	INPU1450
	CALL FCARD(3.0,X,Y,Z,FLAG1,FLAG2,DUM,DUM,DUM)	INPU1460
C	IS IT A COOLANT? -'NO,YES'-	INPU1470
	IF (X.GT.0.0) GO TO 240	INPU1480
C	ASSIGN THE FLOW DIRECTION OF THE COOLANT	INPU1490
	N=-(X-0.1)	INPU1500
	IPATH(N)=Y	INPU1510
C	IS NORMAL FLOW ALLOWABLE? -'NO,YES'-	INPU1520
	IF (ISHAPE.EQ.0.AND.IABS(IPATH(N)).EQ.3) CALL ERROR (6HINPUT2)	INPU1530
C	ASSIGN THE HEIGHT WHEN NORMAL FLOW IS PRESENT	INPU1540
	IF (IABS(IPATH(N)).EQ.3.AND.Z.GT.0.0) HEIGHT=Z/12.0	INPU1550
C	ASSIGN THE COOLANT OR MATERIAL NUMBER	INPU1560
240	MB(K)=X	INPU1570
C=	=====	INPU1580
C	READ THE BLOCK DIMENSIONS	INPU1590
	READ (5,351) RBL(K),RBH(K),ZBL(K),ZBH(K),FLAG1,FLAG2	INPU1600
	CALL FCARD(4.0,RBL(K),RBH(K),ZBL(K),ZBH(K),FLAG1,FLAG2,DUM,DUM)	INPU1610
C=	=====	INPU1620
C	READ THE GAP PROPERTIES	INPU1630
	READ (5,351) RDG(K),X,ZDG(K),Y,FLAG1,FLAG2	INPU1640
	CALL FCARD(4.0,RDG(K),X,ZDG(K),Y,FLAG1,FLAG2,DUM,DUM)	INPU1650
	MGR(K)=X	INPU1660
	MGZ(K)=Y	INPU1670
	IND=0	INPU1680
	IF (X.GT.0.0) IND=IND+1	INPU1690
	IF (Y.GT.0.0) IND=IND+2	INPU1700
C=	=====	INPU1710
C	PRINT THE BLOCK DEFINITION	INPU1720
	IF (IND.EQ.0) GO TO 244	INPU1730
	GO TO (241,242,243), IND	INPU1740
241	WRITE (6,360) K,RBL(K),RBH(K),ZBL(K),ZBH(K),MB(K),RDG(K),MGR(K)	INPU1750
	GO TO 230	INPU1760
242	WRITE (6,370) K,RBL(K),RBH(K),ZBL(K),ZBH(K),MB(K),ZDG(K),MGZ(K)	INPU1770
	GO TO 230	INPU1780
243	WRITE (6,360) K,RBL(K),RBH(K),ZBL(K),ZBH(K),MB(K),RDG(K),MGR(K),ZD	INPU1790
	IG(K),MGZ(K)	INPU1800
	GO TO 230	INPU1810
244	WRITE (6,360) K,RBL(K),RBH(K),ZBL(K),ZBH(K),MB(K)	INPU1820
	GO TO 230	INPU1830

```

C= =====INPU1840
C ASSIGN THE NUMBER OF BLOCKS INPU1850
250 NC=NC+1 INPU1860
CARJ(NC)=0.0 INPU1870
LMAX=K-1 INPU1880
C/ CONVERT THE BLOCK AND GAP DIMENSIONS TO THEIR CORRECT UNITS INPU1890
DO 260 K=1,LMAX INPU1900
IF (RBL(K).LE.0.0) RBL(K)=1.0E-5 INPU1910
RBL(K)=RBL(K)/12.0 INPU1920
IF (RBH(K).LE.0.0) RBH(K)=1.0E-5 INPU1930
RBH(K)=RBH(K)/12.0 INPU1940
ZBL(K)=ZBL(K)/SCALE INPU1950
ZBH(K)=ZBH(K)/SCALE INPU1960
RJD(K)=RJD(K)/12.0 INPU1970
ZD(K)=ZD(K)/SCALE INPU1980
260 CONTINUE INPU1990
CE *****INPU2000
C TEST THAT BLOCK BOUNDARIES COINCIDE WITH GRID LINES INPU2010
DO 261 I=1,IMAX INPU2020
DO 261 J=1,JMAX INPU2030
BTABLE(I,J)=BLANK INPU2040
261 CONTINUE INPU2050
DO 340 K=1,LMAX INPU2060
DO 270 ILG=1,IM INPU2070
IF (RBL(K).GE.0.99999*RL(ILG).AND.RB(K).LE.1.00001*RL(ILG)) GO TO INPU2080
1 280 INPU2090
270 CONTINUE INPU2100
CALL ERROR (6HINPUT3) INPU2110
280 DO 290 IHG=1,IM INPU2120
IF (RBH(K).GE.0.99999*RL(IHG).AND.RBH(K).LE.1.00001*RL(IHG)) GO TO INPU2130
1 300 INPU2140
290 CONTINUE INPU2150
CALL ERROR (6HINPUT4) INPU2160
300 DO 310 JLG=1,JM INPU2170
IF (ZBL(K).GE.0.99999*ZL(JLG).AND.ZB(K).LE.1.00001*ZL(JLG)) GO TO INPU2180
1 320 INPU2190
310 CONTINUE INPU2200
CALL ERROR (6HINPUT5) INPU2210
320 DO 330 JHG=1,JM INPU2220
IF (ZBH(K).GE.0.99999*ZL(JHG).AND.ZBH(K).LE.1.00001*ZL(JHG)) GO TO INPU2230
1 331 INPU2240
330 CONTINUE INPU2250
CALL ERROR (6HINPUT6) INPU2260
331 IF (MB(K).GT.0) GO TO 334 INPU2270
DO 332 J=JLG,JHG INPU2280
BTABLE(ILG,J)=STAR INPU2290

```

```

BTABLE(IHG,J)=STAR                                INPU2300
332 CONTINUE                                         INPU2310
DO 333 I=ILG,IHG                                    INPU2320
BTABLE(I,JLG)=STAR                                  INPU2330
BTABLE(I,JHG)=STAR                                  INPU2340
333 CONTINUE                                         INPU2350
GO TO 340                                             INPU2360
334 RHS=JOT                                          INPU2370
ZHS=JOT                                              INPU2380
IF (RUG(K).GT.0.0) RHS=GAP                          INPU2390
IF (ZUG(K).GT.0.0) ZHS=GAP                          INPU2400
DO 335 J=JLG,JHG                                    INPU2410
IF (BTABLE(IHG,J).NE.STAR .AND. BTABLE(I,JHG).NE.GAP) BTABLE(I,J)=RHS INPU2420
Z)=RHS                                              INPU2430
335 CONTINUE                                         INPU2440
DO 336 I=ILG,IHG                                    INPU2450
IF (BTABLE(I,JHG).NE.STAR .AND. BTABLE(I,JLG).NE.GAP) BTABLE(I,J)=ZHS INPU2460
Z)=ZHS                                              INPU2470
336 CONTINUE                                         INPU2480
340 CONTINUE                                         INPU2490
WRITE (6,380) (ZA(I),I=1,12)                       INPU2500
380 FORMAT(141,30X,12A6,///,50X,17HBOUNDARY OVERLAY,/,55X,29H* WHERE INPU2510
COOLANTS ARE PRESENT,/,55X,25H* WHERE GAPS ARE PRESENT,/,55X,37 INPU2520
H. WHERE GAPS OR COOLANTS NOT PRESENT,/)           INPU2530
CALL BARRAY                                          INPU2540
C= =====INPU2550
RETURN                                              INPU2560
C= =====INPU2570
350 FORMAT (3E12.4,3GX,A6,AZ)                      INPU2580
351 FORMAT (4E12.4,2GX,A6,AZ)                      INPU2590
360 FORMAT (17,4F14.4,I9,F14.4,I7,F13.4,I7)        INPU2600
370 FORMAT (17,4F14.4,I9,21X,F13.4,I7)              INPU2610
C= =====INPU2620
CE *****INPU2630
CE *****INPU2640
CE *****INPU2650
SUBROUTINE TITLE                                     INPU2660
C= =====INPU2670
C READ AND PRINT TITLE CARDS UNTIL A BLANK CARD    INPU2680
C IS ENCOUNTERED. PRESERVE THE FIRST LINE         INPU2690
C= =====INPU2700
INTEGER ZZ (14)                                     INPU2710
C= =====INPU2720
C READ, RECORD AND PRINT THE FIRST TITLE CARD     INPU2730
READ (5,140) (ZA(I),I=1,14)                       INPU2740
CALL ACARD(ZA)                                       INPU2750

```

	WRITE (6,150) (ZA(I),I=1,12)	INPU2760
C	READ THE NEXT TITLE CARD	INPU2770
100	READ (5,140) (ZZ(I),I=1,14)	INPU2780
C/	HAVE ALL TITLE CARDS BEEN READ? -'NO,YES'-	INPU2790
	DO 110 I=1,12	INPU2800
	IF (ZZ(I).NE.BLANKS) GO TO 130	INPU2810
110	CONTINUE	INPU2820
	NC=NC+1	INPU2830
	CARD(NC)=0.0	INPU2840
	WRITE (6,120)	INPU2850
120	FORMAT (1H0,15X,7HOPTIONS)	INPU2860
CE	*****	INPU2870
	RETURN	INPU2880
C=	=====	INPU2890
C	PRINT THE TITLE CARD	INPU2900
130	WRITE (6,150) (ZZ(I),I=1,12)	INPU2910
	CALL ACARD(ZZ)	INPU2920
	GO TO 100	INPU2930
C=	=====	INPU2940
140	FORMAT (13A6,A2)	INPU2950
150	FORMAT (10X,12A6)	INPU2960
C=	=====	INPU2970
CE	*****	INPU2980
CE	*****	INPU2990
CE	*****	INPU3000
	SUBROUTINE GRID (RL1,IM1)	INPU3010
C=	=====	INPU3020
C	READ ONE SET OF GRID DATA CARDS	INPU3030
C=	=====	INPU3040
C	ERROR STOPS=	INPU3050
C	GRID1 THERE ARE MORE GRIDLINES THEN ALLOWED IN ONE OF	INPU3060
C	THE DIMENSIONS. A BLANK CARD HAS BEEN LEFT OUT.	INPU3070
C	=====	INPU3080
C	=====	INPU3090
	INTEGER IM	INPU3100
	DIMENSION RL (8),RL1 (1)	INPU3110
C=	=====	INPU3120
	ILS=1	INPU3130
	IHS=6	INPU3140
C	READ THE NEXT GRID DATA CARD	INPU3150
100	READ (5,140) (RL(I),I=1,8)	INPU3160
C	HAVE ALL GRID DATA CARDS BEEN READ? -'YES,NO'-	INPU3170
	IF (RL(1).EQ.0.0.AND.RL(2).EQ.0.0) GO TO 120	INPU3180
	CALL FCARD(6.0,RL(1),RL(2),RL(3),RL(4),RL(5),RL(6),RL(7),RL(8))	INPU3190
C=	=====	INPU3200
	IM=MIN0(IHS,IM1)	INPU3210

```

C      ARE THERE TOO MANY GRID DATA CARDS ? -'YES,NO'- .      INPU3220
C      PRINT ERROR GRID 1      INPU3230
C      IF (ILS.GT.IM) CALL ERROR (GM GRID1)      INPU3240
C      RECORD THE CONTENTS OF THE CURRENT GRID DATA CARD      INPU3250
C      DO 110 I=ILS,IM      INPU3260
C      J=I+1-ILS      INPU3270
C      RL1(I)=RL(J)      INPU3280
110  CONTINUE      INPU3290
C      PREPARE FOR THE READING OF THE NEXT GRID DATA CARD      INPU3300
C      ILS=IHS+1      INPU3310
C      IHS=IHS+6      INPU3320
C      GO TO 100      INPU3330
C= =====INPU3340
C/ DETERMINE THE NUMBER OF GRID LINES READ      INPU3350
120  NC=NC+1      INPU3360
C      CARD(NC)=0.00      INPU3370
C      IM=ILS      INPU3380
130  IM=IM-1      INPU3390
C      IF (RL1(IM).LE.0.0) GO TO 130      INPU3400
C      IM=IM      INPU3410
CE *****INPU3420
C      RETURN      INPU3430
C= =====INPU3440
140  FORMAT (B12.4,A6,A2)      INPU3450
CE *****INPU3460
C= =====INPU3470
CE *****INPU3480
CE *****INPU3490
CE *****INPU3500
C      SUBROUTINE BARRAY      INPU3510
C= =====INPU3520
C      DIMENSION      NUMBER( 60)      INPU3530
C      DATA NUMBER      /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19, INPU3540
1      20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35, INPU3550
2      36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51, INPU3560
3      52,53,54,55,56,57,58,59,60/ INPU3570
C= =====INPU3580
C      WRITE THE PROPER HEADING      INPU3590
100  WRITE (6,150) PNAME(1),PNAME(2)      INPU3600
C      NUM2=24      INPU3610
C      IF (NUM2.GE.IM) NUM2=IM      INPU3620
C      WRITE (6,160) (NUMBER(L),L=1,NUM2)      INPU3630
C      WRITE (6,170)      INPU3640
C      DO 110 JJ=1,JM      INPU3650
C      J=JJ      INPU3660
C      IF (JW(21)) J=JM-JJ+1      INPU3670

```

```

110 WRITE (6,180) J,(BTABLE(I,J),I=1,NUM2)
GO TO 140
130 NUM1=NUM2+1
NUM2=NUM2+25
IF (NUM2.GE.IM) NUM2=IM
WRITE (6,190) (NUMBER(L),L=NUM1,NUM2)
WRITE (6,170)
DO 120 JJ=1,JM
J=JJ
IF (SW(21)) J=JM-JJ+1
120 WRITE (6,200) J,(BTABLE(I,J),I=NUM1,NUM2)
140 IF (NUM2.GE.IM) RETURN
GO TO 130
C= =====
150 FORMAT (1H0,42X,4HTHE ,A6,29I(I) GRID LINES ARE HORIZONTAL/43X,4HTHE ,A6,27H(J) GRID LINES ARE VERTICAL)
160 FORMAT(1H0,6X,25I5)
170 FORMAT (1H0)
180 FORMAT (14,7X,24A5)
190 FORMAT(1H0,1X,26I5)
200 FORMAT(14,2X,25A5)
CE *****
CE *****
SUBROUTINE ACARD(AC)
C= =====
JIMENSION 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 *****
CE *****
END

```

```

INPU3680
INPU3690
INPU3700
INPU3710
INPU3720
INPU3730
INPU3740
INPU3750
INPU3760
INPU3770
INPU3780
INPU3790
INPU3800
INPU3810
INPU3820
INPU3830
INPU3840
INPU3850
INPU3860
INPU3870
INPU3880
INPU3890
INPU3900
INPU3910
INPU3920
INPU3930
INPU3940
INPU3950
INPU3960
INPU3970
INPU3980
INPU3990
INPU4000
INPU4010
INPU4020
INPU4030
INPU4040
INPU4050

```

```

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

```

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

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


```

SUBROUTINE CHECK                                CHFC 10
INCLUDE COMDIM                                CHFC 20
C=====CHFC 30
C CHECK THE GEOMETRY DATA                      CHFC 40
C=====CHFC 50
C ERROR STOPS=                                CHFC 60
C CHECK1 THERE ARE TOO MANY RADIAL-X POINTS AND GRID LINES CHFC 70
C CHECK2 THERE ARE TOO MANY AXIAL-Y-THETA POINTS AND GRID LINESCHFC 80
C CHECK3 THE RADIAL-X GRID DATA IS OUT OF ORDER CHFC 90
C CHECK5 THE AXIAL-Y-THETA GRID DATA IS OUT OF ORDER CHFC 100
C CHECK7 THERE ARE TOO MANY BLOCKS. CHFC 110
C CHECK8 THE LOW RADIAL-X BOUNDARY IS LARGER THAN THE HIGH CHFC 120
C RADIAL-X BOUNDARY FOR SOME BLOCK CHFC 130
C CHECK9 THE LOW AXIAL-Y-THETA BOUNDARY IS GREATER THAN THE CHFC 140
C HIGH AXIAL-Y-THETA BOUNDARY FOR SOME BLOCK CHFC 150
C CHECK11 THE MATERIAL NUMBER FOR A BLOCK IS LARGER THAN THE CHFC 160
C MAXIMUM NUMBER OF MATERIALS AND COOLANTS ALLOWED. CHFC 170
C CHECK12 A RADIAL-X GAP MATERIAL NUMBER IS TOO HIGH CHFC 180
C CHECK13 AN AXIAL-Y-THETA GAP MATERIAL NUMBER IS TOO HIGH CHFC 190
C=====CHFC 200
C=====CHFC 210
C ARE THERE TOO MANY RADIAL-X GRIDLINES? -'YES,NO'- CHFC 220
C PRINT ERROR CHECK1 CHFC 230
C IF (IMAX.GT.MAXRP) CALL ERROR (6HCHECK1) CHFC 240
C ARE THERE TOO MANY AXIAL-Y-THETA GRIDLINES? -'YES,NO'- CHFC 250
C PRINT ERROR CHECK2 CHFC 260
C IF (JMAX.GT.MAXZP) CALL ERROR (6HCHECK2) CHFC 270
C=====CHFC 280
C/ CHECK TO SEE THAT THE RADIAL-X GRID LINES ARE IN AN ASCENDING CHFC 290
C -ORDER CHFC 300
C DO 100 I=2,IM CHFC 310
C IF (RL(I).LE.RL(I-1)) CALL ERROR (6HCHECK4) CHFC 320
100 CONTINUE CHFC 330
C=====CHFC 340
C/ CHECK TO SEE THAT THE AXIAL-Y-THETA GRID LINES ARE IN AN ASCENDINGCHFC 350
C -ORDER CHFC 360
C DO 110 J=2,J1 CHFC 370
C IF (ZL(J).LE.ZL(J-1)) CALL ERROR (6HCHECK5) CHFC 380
110 CONTINUE CHFC 390
C=====CHFC 400
C ARE THERE TOO MANY BLOCKS? -'YES,NO'- CHFC 410
C PRINT ERROR CHECK7 CHFC 420
C IF (LMAX.GT.MAXNB) CALL ERROR (6HCHECK7) CHFC 430
C=====CHFC 440
CB CHECK THE BLOCK GEOMETRY FOR INTERNAL CONSISTENCY CHFC 450

```

C	CHECK EVERY BLOCK IN TURN	CHFC 460
	DO 120 L=1,LMAX	CHFC 470
C	ARE THE RADIAL-X BOUNDARIES IN ORDER? -'NO,YES'-	CHFC 480
C	PRINT ERROR CHECK8	CHFC 490
	IF (RBL(L).GT.RBH(L)) CALL ERROR (6H,CHECK8)	CHFC 500
C	ARE THE AXIAL-Y-THETA BOUNDARIES IN ORDER? -'NO,YES'-	CHFC 510
C	PRINT ERROR CHECK9	CHFC 520
	IF (ZBL(L).GT.ZBH(L)) CALL ERROR (6H,CHECK9)	CHFC 530
C	IS THE MATERIAL NUMBER A VALID ONE? -'NO,YES'-	CHFC 540
C	PRINT ERROR CHECK11	CHFC 550
	IF (IABS(MB(L)).GT.MAXMAT) CALL ERROR (7H,CHECK11)	CHFC 560
C	IS THE MATERIAL NUMBER OF THE RADIAL-X GAP A VALID ONE? -'NO,YES'-	CHFC 570
C	PRINT ERROR CHECK12	CHFC 580
	IF (MGR(L).GT.MAXMAT) CALL ERROR (7H,CHECK12)	CHFC 590
C	IS THE MATERIAL NUMBER OF THE AXIAL-Y-THETA GAP A VALID ONE? -'NO,YES'-	CHFC 600
C	PRINT ERROR CHECK13	CHFC 610
	IF (MGZ(L).GT.MAXMAT) CALL ERROR (7H,CHECK13)	CHFC 620
C	HAVE ALL BLOCKS BEEN CHECKED ? -'YES,NO'-	CHFC 630
120	CONTINUE	CHFC 640
CE	*****	CHEC 650
	RETURN	CHFC 660
C=	=====	CHFC 670
	END	CHFC 680

```

SUBROUTINE POINTS                                POINT 10
INCLUDE COMDIM                                  POINT 20
C= =====POINT 30
C ASSIGN THE POINTS, BLOCKS AND GAPS              POINT 40
C= =====POINT 50
C ERROR STOPS=                                  POINT 60
C   POINTS1 A PART OF THE SYSTEM WAS NOT DESCRIBED BY ANY BLOCK. POINT 70
C   POINTS2 THERE ARE TOO MANY RADIAL-X GAPS.      POINT 80
C   POINTS3 THERE ARE TOO MANY AXIAL-Y-THETA GAPS. POINT 90
C   POINTS5 A PART OF THE SYSTEM HAS BEEN DESCRIBED BY MORE THAN POINT 100
C           ONE BLOCK.                             POINT 110
C   POINTS6 NO FLOW DIRECTION HAS BEEN ASSIGNED FOR SOME COOLANT. POINT 120
C   POINTS7 1. AN EXTERNAL COOLANT IS FLOWING INTO A RADIAL-X POINT 130
C             BOUNDARY, OR                          POINT 140
C           2. AN INTERNAL RADIAL-X FLOW COOLANT BLOCK IS NOT POINT 150
C             TRAVERSED BY AT LEAST ONE RADIAL-X GRID LINE. POINT 160
C   POINTS8 1. AN EXTERNAL COOLANT IS FLOWING INTO AN AXIAL-Y- POINT 170
C             THETA BOUNDARY OR                     POINT 180
C           2. AN INTERNAL AXIAL-Y-THETA FLOW COOLANT BLOCK IS POINT 190
C             NOT TRAVERSED BY AT LEAST ONE AXIAL-Y-THETA POINT 200
C             GRID LINE.                           POINT 210
C   POINTS9 A GAP HAS BEEN SPECIFIED ON THE HIGH RADIAL-X POINT 220
C            BOUNDARY OF A COOLANT.                 POINT 230
C   POINTS10 A GAP HAS BEEN SPECIFIED ON THE LOW RADIAL-X POINT 240
C            BOUNDARY OF A COOLANT.                 POINT 250
C   POINTS11 A GAP HAS BEEN SPECIFIED ON THE HIGH AXIAL-Y-THETA POINT 260
C            BOUNDARY OF A COOLANT.                 POINT 270
C   POINTS12 A GAP HAS BEEN SPECIFIED ON THE LOW AXIAL-Y-THETA POINT 280
C            BOUNDARY OF A COOLANT.                 POINT 290
C =====POINT 300
C= =====POINT 310
C INITIALIZE THE BLOCK COMPLETENESS AND BOUNDARY OVERLAY TABLES POINT 320
C   DO 80 I=1,IMAX                                POINT 330
C     DO 80 J=1,JMAX                                POINT 340
C     MT(I,J)=0                                     POINT 350
C 80 CONTINUE                                       POINT 360
C     MT(1,1)=1                                    POINT 370
C     MT(1,JMAX)=1                                 POINT 380
C     MT(IMAX,1)=1                                 POINT 390
C     MT(IMAX,JMAX)=1                              POINT 400
C= =====POINT 410
C ASSIGN THE RADIAL-X BLOCKS AND FIND THE RADIAL-X GAPS POINT 420
C CALL BOUND (RL,RP,1M,IGR,RBL,RBH,RDQ,IL,IH,NRG,1) POINT 430
C ARE THERE TOO MANY RADIAL-X GAPS                POINT 440
C IF (NRG.GT.MAXRG) CALL ERROR (7HP0INTS2)         POINT 450

```

```

C= =====POIN 460
C   ASSIGN THE AXIAL-Y-THETA BLOCKS AND FIND THE AXIAL-Y-THETA GAPS POIN 470
C   CALL BOUNDA (ZL,ZP,JM,JGZ,ZBL,ZBH,ZDR,JL,JH,NZG,2) POIN 480
C   ARE THERE TOO MANY AXIAL-Y-THETA GAPS POIN 490
C   IF (NZG.GT.MAXZG) CALL ERROR (7HPOINTS3) POIN 500
C= =====POIN 510
C   INITIALIZE ARRAYS POIN 520
C   DO 90 IG=1,MAXRG POIN 530
C   DO 90 J=1,MAXZP POIN 540
C   MATRG(IG,J)=0 POIN 550
C   RBBIL(IG,J)=0.0 POIN 560
C   RBBIH(IG,J)=0.0 POIN 570
C   GAPR(IG,J)=0.0 POIN 580
C   90 CONTINUE POIN 590
C   DO 100 JG=1,MAXZG POIN 600
C   DO 100 I=1,MAXRP POIN 610
C   MATZG(JG,I)=0 POIN 620
C   ZBBIL(JG,I)=0.0 POIN 630
C   ZBBIH(JG,I)=0.0 POIN 640
C   GAPZ(JG,I)=0.0 POIN 650
C   100 CONTINUE POIN 660
C= =====POIN 670
C   LOOK AT EVERY BLOCK POIN 680
C   DO 230 L=1,LMAX POIN 690
C   ASSIGN THE CORNERS OF THE CURRENT BLOCK POIN 700
C   ILS=IL(L) POIN 710
C   IHS=IH(L) POIN 720
C   JLS=JL(L) POIN 730
C   JHS=JH(L) POIN 740
C   IS THIS BLOCK A COOLANT OR A SOLID MATERIAL? -'COOL.,SOLID'- POIN 750
C   IF (MB(L).LE.0) GO TO 160 POIN 760
C= =====POIN 770
C   IS A RADIAL-X GAP PRESENT? -'NO,YES'- POIN 780
C   IF (RDG(L).LE.0.) GO TO 120 POIN 790
C= =====POIN 800
C   ASSIGN THE RADIAL-X GAP POIN 810
C   IG=1GR(IHS) POIN 820
C   DO 110 J=JLS,JHS POIN 830
C   MATRG(IG,J)=MGR(L) POIN 840
C   IF (GAPR(IG,J).GE.0.0) GO TO 104 POIN 850
C   CALL ERROR(8HPOINTS10) POIN 860
C   GO TO 140 POIN 870
C   104 GAPR(IG,J)=RDG(L) POIN 880
C   RBBIL(IG,J)=460.0 POIN 890
C   RBBIH(IG,J)=460.0 POIN 900
C   110 CONTINUE POIN 910

```

```

C= =====POINTN 920
C   IS AN AXIAL-Y-THETA GAP PRESENT? -'NO,YES'-          POINTN 930
120 IF (ZUG(L).LE.0.0) GO TO 140                          POINTN 940
C= =====POINTN 950
C   ASSIGN THE AXIAL-Y-THETA GAP                          POINTN 960
   JG=JGZ(JHS)                                           POINTN 970
   DO 130 I=ILS,IHS                                     POINTN 980
   MATZG(JG,I)=MGZ(L)                                    POINTN 990
   IF(GAPZ(JG,I).GE.0.0)GO TO 124                        POINTN1000
   CALL ERROR(6HPOINTS12)                                POINTN1010
   GO TO 140                                              POINTN1020
124 GAPZ(JG,I)=ZUG(L)                                    POINTN1030
   ZBSTL(JG,I)=450.0                                    POINTN1040
   ZBSTH(JG,I)=460.0                                    POINTN1050
130 CONTINUE                                             POINTN1060
C= =====POINTN1070
C/  RECORD ALL POINTS BELONGING TO THE CURRENT BLOCK -'IMPOS.,OK'- POINTN1080
140 DO 150 I=ILS,IHS                                    POINTN1090
   DO 150 J=JLS,JHS                                     POINTN1100
   IF (MT(I,J).NE.0) CALL ERROR (7HPOINTS55)            POINTN1110
   MT(I,J)=1                                             POINTN1120
150 CONTINUE                                             POINTN1130
   GO TO 200                                              POINTN1140
CE *****POINTN1150
C= =====POINTN1160
C   ASSIGN THE COOLANT NUMBER                             POINTN1170
160 J=IAHS(MB(L))                                       POINTN1180
C   ASSIGN THE FLOW DIRECTION                             POINTN1190
   IP=IABS(IPATH(J))                                    POINTN1200
C   IS IT A VALID DIRECTION? -'YES,NO'-                 POINTN1210
   IF (IP.GT.0.AND.IP.LT.4) GO TO 170                   POINTN1220
C   IMPOSSIBLE FLOW DIRECTION NUMBER                     POINTN1230
   CALL ERROR (7HPOINTS6)                               POINTN1240
   GO TO 140                                              POINTN1250
C   IS THE FLOW IN THE RADIAL DIRECTION? -'YES,NO'-     POINTN1260
170 IF (IP.EQ.1) GO TO 200                               POINTN1270
C   SET TWO SWITCHES TO INDICATE ON WHICH SIDE OF ITS RADIAL POINTN1280
C   .BOUNDARIES THE COOLANT LIES                         POINTN1290
   IG1=IGR(IHS)                                         POINTN1300
   IF(1LS.LE.1)GO TO 172                                POINTN1310
   IG2=IGR(1LS-1)                                       POINTN1320
   GO TO 174                                             POINTN1330
172 IG2=0                                                POINTN1340
174 DO 190 J=JLS,JHS                                    POINTN1350
   IF(IG2.LE.0)GO TO 180                                POINTN1360
   IF(GAPR(IG2,J).LE.0.0)GO TO 176                     POINTN1370

```

CALL ERROR(8HPOINTS10)	POINT1380
GO TO 140	POINT1390
176 GAPR(IG2,J)=-1.0E-10+GAPR(IG2,J)	POINT1400
RBBIL(IG2,J)=460.0	POINT1410
RBBIH(IG2,J)=460.0	POINT1420
180 IF(IG1.LE.0)GO TO 190	POINT1430
IF(GAPR(IG1,J).LE.0.0)GO TO 184	POINT1440
CALL ERROR(7HPOINTS9)	POINT1450
GO TO 140	POINT1460
184 GAPR(IG1,J)=-2.0E-10+GAPR(IG1,J)	POINT1470
RBBIL(IG1,J)=460.0	POINT1480
RBBIH(IG1,J)=460.0	POINT1490
190 CONTINUE	POINT1500
C CAN THE FLOW BE IN THE AXIAL DIRECTION? -'NO,YES'-	POINT1510
C WRONG FLOW DIRECTION WAS SELECTED	POINT1520
IF(JLS.EQ.JHS.AND.IP.NE.3) CALL ERROR(7HPOINTS8)	POINT1530
C IS THE FLOW IN THE AXIAL DIRECTION? -'YES,NO'-	POINT1540
200 IF(IP.EQ.2) GO TO 140	POINT1550
C SET TWO SWITCHES TO INDICATE ON WHICH SIDE OF ITS AXIAL	POINT1560
C .300J JARIES THE COOLANT LIES	POINT1570
JG1=JGZ(JHS)	POINT1580
IF(JLS.LE.1)GO TO 202	POINT1590
JG2=JGZ(JLS-1)	POINT1600
GO TO 204	POINT1610
202 JG2=0	POINT1620
204 DO 220 I=ILS,IHS	POINT1630
IF(JG1.LE.0)GO TO 210	POINT1640
IF(GAPZ(JG1,I).LE.0.0)GO TO 206	POINT1650
CALL ERROR(8HPOINTS11)	POINT1660
GO TO 140	POINT1670
206 GAPZ(JG1,I)=-2.0E-10+GAPZ(JG1,I)	POINT1680
ZBBIL(JG1,I)=460.0	POINT1690
ZBBIH(JG1,I)=460.0	POINT1700
210 IF(JG2.LE.0)GO TO 220	POINT1710
IF(GAPZ(JG2,I).LE.0)GO TO 214	POINT1720
CALL ERROR(8HPOINTS12)	POINT1730
GO TO 140	POINT1740
214 GAPZ(JG2,I)=-1.0E-10+GAPZ(JG2,I)	POINT1750
ZBBIL(JG2,I)=460.0	POINT1760
ZBBIH(JG2,I)=460.0	POINT1770
220 CONTINUE	POINT1780
C CAN THE FLOW BE IN THE RADIAL DIRECTION? -'NO,YES'-	POINT1790
C WRONG FLOW DIRECTION WAS SELECTED	POINT1800
IF(JLS.EQ.IHS.AND.IP.NE.3) CALL ERROR(7HPOINTS7)	POINT1810
GO TO 140	POINT1820
C= =====	POINT1830

```

C      HAVE ALL BLOCKS BEEN CHECKED? -'YES,NOTYET'-                      POINT1840
230 CONTINUE                                                                POINT1850
C= =====                                                                POINT1860
C      INITIALIZE THE MATRG AND MATZG VALUES WHICH ARE ON GAPLINES      POINT1870
C      BOUNDING COULANTS.                                                POINT1880
      DO 233 I=1,NRG                                                       POINT1890
      DO 233 J=1,JMAX                                                       POINT1900
233 IF(GAPR(I,J).LT.(-.5E-10)) MATRG(I,J)=100                             POINT1910
      DO 237 I=1,NZG                                                       POINT1920
      DO 237 J=1,JMAX                                                       POINT1930
237 IF(GAPZ(I,J).LT.(-.5E-10)) MATZG(I,J)=100                             POINT1940
C= =====                                                                POINT1950
C/      HAVE ALL POINTS BEEN ASSIGNED TO A BLOCK? -'NO,YES'-            POINT1960
      DO 239 I=1,JMAX                                                       POINT1970
      DO 240 J=1,JMAX                                                       POINT1980
      IF (MT(I,J).LE.0) CALL ERROR (7HPOINTS1)                             POINT1990
240 CONTINUE                                                                POINT2000
250 CONTINUE                                                                POINT2010
CE      *****POINT2020
      RETURN                                                                POINT2030
C= =====                                                                POINT2040
      END                                                                    POINT2050

```

```

SUBROUTINE BOUNDX (AL,AP,M,IGA,ABL,ABH,ADG,ILA,IHA,NAG,IP)      BOJN 10
INCLUDE COMJIM                                                  BOJN 20
C=====BOJN 30
C ASSIGN THE BLOCK AND GAPLINE LOCATIONS FOR ONE DIMENSION      BOJN 40
C=====BOJN 50
C ERROR STOPS=                                                  BOJN 60
C   BOUNDA1 THE LOW BOUNDARY FOR A BLOCK IS TOO LARGE OR        BOJN 70
C           LARGER THAN THE LARGEST BOUNDARY DEFINED IN THAT     BOJN 80
C           DIMENSION.                                           BOJN 90
C   BOUNDA2 THE HIGH BOUNDARY OF A BLOCK IS LARGER THAN THE     BOJN 100
C           LARGEST BOUNDARY DEFINED IN THAT DIMENSION.         BOJN 110
C   BOUNDA3 AN INTERNAL COOLANT BLOCK HAS BEEN SPECIFIED IN     BOJN 120
C           VIOLATION OF CODE LIMITATIONS. MOST LIKELY ERRORS AREBOJN 130
C           1. A NORMAL COOLANT HAS BEEN SPECIFIED FOR A        BOJN 140
C           PROBLEM IN CYLINDRICAL COORDINATES.                BOJN 150
C           2. AN INTERNAL COOLANT HAS TWO BOUNDARIES WHICH     BOJN 160
C           ARE COINCIDENT.                                     BOJN 170
C=====BOJN 180
C=====BOJN 190
C DIMENSION AL ( 1),AP ( 1),IGA ( 1),ABL ( 1),ABH ( 1),ADG ( 1),ILA ( 1),IHA ( 1) BOJN 200
1      ABH ( 1),ADG ( 1),ILA ( 1),IHA ( 1) BOJN 210
C=====BOJN 220
C CALCULATE THE POINT LOCATIONS BOJN 230
C AP(1)=AL(1) BOJN 240
C DO 100 I=2,M BOJN 250
C AP(I)=(AL(I)+AL(I-1))*0.5 BOJN 260
100 CONTINUE BOJN 270
C AP(M+1)=AL(M) BOJN 280
C=====BOJN 290
C ERASE THE GAP LOCATION ARRAY BOJN 300
C DO 110 I=1,M BOJN 310
C IGA(I)=0 BOJN 320
110 CONTINUE BOJN 330
C M1=M+1 BOJN 340
C=====BOJN 350
C FIND THE BLOCK BOUNDARY INDICES IN THE CURRENT DIMENSION BOJN 360
C FOR EACH BLOCK BOJN 370
C DO 190 L=1,LMAX BOJN 380
C IS IT A COOLANT OR A SOLID MATERIAL BLOCK? -'SOLID,COOL.-' BOJN 390
C IF (MB(L).GT.0) GO TO 140 BOJN 400
C IS IT AN EXTERNAL OR INTERNAL COOLANT? -'INTERN,EXTERN'- BOJN 410
C IF (ABL(L).NE.ABH(L)) GO TO 140 BOJN 420
CB ASSIGN THE EXTERNAL COOLANT BLOCK BOJN 430
C IS IT ON THE LOW INDEX OUTSIDE -'YES,NO'- BOJN 440
C IF (ABL(L).EQ.AP(1)) GO TO 120 BOJN 450

```


C	IS IT ON THE HIGH INDEX OUTSIDE -'YES,NO'-	BOJN 460
	IF (ABL(L).EQ.AP(M1)) GO TO 130	BOJN 470
C	ILLEGAL INTERNAL COOLANT CHANNEL	BOJN 480
	CALL ERROR (7HBOUND3)	BOJN 490
C	ASSIGN THE LOW INDEX OUTSIDE COOLANT LIMITS	BOJN 500
120	ILA(L)=1	BOJN 510
	IHA(L)=1	BOJN 520
	IGA(1)=-1	BOJN 530
	GO TO 190	BOJN 540
C	ASSIGN THE HIGH INDEX OUTSIDE COOLANT LIMITS	BOJN 550
130	ILA(L)=M1	BOJN 560
	IHA(L)=M1	BOJN 570
	IGA(M)=-1	BOJN 580
	GO TO 190	BOJN 590
CE	*****	BOJN 600
CB	ASSIGN THE SOLID MATERIAL OR INTERNAL COOLANT BLOCK	BOJN 610
C/	FINJ THE LOW BLOCK BOUNDARY INDEX -'OK,NONE'-	BOJN 620
140	DO 150 I=2,M	BOJN 630
	IF (ABL(L).LT.AP(I)) GO TO 160	BOJN 640
150	CONTINUE	BOJN 650
CE	*****	BOJN 660
C	THE LOW BLOCK BOUNDARY LIES OUTSIDE THE SYSTEM	BOJN 670
	CALL ERROR (7HBOUND1)	BOJN 680
C=	=====	BOJN 690
C	ASSIGN THE LOW BLOCK BOUNDARY	BOJN 700
160	ILA(L)=1	BOJN 710
C/	FINJ THE HIGH BLOCK BOUNDARY INDEX -'OK,NONE'-	BOJN 720
	I=I+1	BOJN 730
	DO 170 J=I,M1	BOJN 740
	IF (ABH(L).LE.AP(J)) GO TO 180	BOJN 750
170	CONTINUE	BOJN 760
CE	*****	BOJN 770
C	THE HIGH BLOCK BOUNDARY LIES OUTSIDE THE SYSTEM	BOJN 780
	CALL ERROR (7HBOUND2)	BOJN 790
C	ASSIGN THE HIGH BLOCK BOUNDARY INDEX	BOJN 800
180	IHA(L)=J-1	BOJN 810
C=	=====	BOJN 820
C	SET THE GAP SWITCH WHEN NECESSARY	BOJN 830
	IF (ADG(L).GT.0.0) IGA(J-1)=-1	BOJN 840
C	IS IT A COOLANT BLOCK -'NO,YES'-	BOJN 850
	IF (MB(L).GT.0) GO TO 190	BOJN 860
	N=-MB(L)	BOJN 870
C	DOES THE COOLANT FLOW IN THE CURRENT DIRECTION -'YES,NO'-	BOJN 880
	IF (IP.EQ.IAUS(IPATH(N))) GO TO 190	BOJN 890
C	ASSIGN A GAP BOUNDARY ON BOTH SIDES OF THE COOLANT	BOJN 900
	IGA(J-1)=-1	BOJN 910

	IGA(1-2)=-1	BOIJN 920
C	HAVE ALL BLOCKS BEEN CHECKED? -'YES,NO'-	BOIJN 930
190	CONTINUE	BOIJN 940
C=	=====	BOIJN 950
C	INSPECT THE GAP INDICES AND ASSIGN THE BOUNDARY NUMBERS	BOIJN 960
	J=0	BOIJN 970
	DO 200 I=1,M	BOIJN 980
	IF (IGA(I).EQ.0) GO TO 200	BOIJN 990
	J=J+1	BOIJN1000
	IGA(1)=J	BOIJN1010
200	CONTINUE	BOIJN1020
	NAG=J	BOIJN1030
CE	*****	BOIJN1040
CE	*****	BOIJN1050
	RETURN	BOIJN1060
C=	=====	BOIJN1070
	END	BOIJN1080

	SUBROUTINE CONSTA	CONS 10
	INCLUDE COMQIM	CONS 20
C=	=====	CONS 30
C	CALCULATE THE CORRECTION CONSTANTS	CONS 40
C=	=====	CONS 50
C=	=====	CONS 60
C/	CALCULATE RADIAL GEOMETRIC CORRECTION FACTOR 1 (RLN TABLE C1)	CONS 70
	DO 110 I=1,IM	CONS 80
	DEL1(I)=RP(I+1)-RP(I)	CONS 90
	IF (ISHAPE.EQ.1) GO TO 100	CONS 100
	RLN(I)=ALOG(RL(I)/RP(I))/ALOG(RP(I+1)/RP(I))	CONS 110
	GO TO 110	CONS 120
	100 RLN(I)=(RL(I)-RP(I))/DEL1(I)	CONS 130
	110 CONTINUE	CONS 140
CE	*****	CONS 150
C	CALCULATE AXIAL GEOMETRIC CORRECTION FACTOR 1 (ZLN TABLE C1)	CONS 160
	DO 120 J=1,JM	CONS 170
	DEL2(J)=ZP(J+1)-ZP(J)	CONS 180
	ZLN(J)=(ZL(J)-ZP(J))/DEL2(J)	CONS 190
	120 CONTINUE	CONS 200
C/	CALCULATE RADIAL GEOMETRIC GAP CORRECTION FACTORS 2 AND 3	CONS 210
C	.(RATIOK,RATIOH, TABLES C2,C4)	CONS 220
	DO 140 I=1,IM1	CONS 230
	IF (IGR(I).LE.0) GO TO 140	CONS 240
	IG=1, R(I)	CONS 250
	IF (ISHAPE.EQ.1) GO TO 130	CONS 260
	RATIOK(IG)=ALOG(RP(I+1)/RP(I))/ALOG(RP(I+1)/RL(I))	CONS 270
	RATIOH(IG)=RL(I)*ALOG(RP(I+1)/RL(I))	CONS 280
	RATIOC(IG)=1.0/(RL(I)*ALOG(RP(I+1)/RP(I)))	CONS 290
	GO TO 140	CONS 300
	130 RATIOK(IG)=DEL1(I)/(RP(I+1)-RL(I))	CONS 310
	RATIOH(IG)=RP(I+1)-RL(I)	CONS 320
	RATIOC(IG)=1.0/DEL1(I)	CONS 330
	140 CONTINUE	CONS 340
	RATIOK(VRG)=0.	CONS 350
	RATIOH(VRG)=0.	CONS 360
	IF (ISHAPE.EQ.1) GO TO 143	CONS 370
	RATIOC(VRG)=1.0/(RL(IM)*ALOG(RP(IM+1)/RP(IM)))	CONS 380
	GO TO 145	CONS 390
	143 RATIOC(VRG)=1.0/DEL1(IM)	CONS 400
	145 CONTINUE	CONS 410
CE	*****	CONS 420
C/	CALCULATE AXIAL GEOMETRIC GAP CORRECTION FACTORS 2 AND 3	CONS 430
C	.(ZATIOK,ZATIOH, TABLES C2,C4,	CONS 440
	DO 150 J=1,JM1	CONS 450

	IF (JGZ(J).LE.0) GO TO 150	CONS 460
	JG=JGZ(J)	CONS 470
	ZATIOK(JG)=DELZ(J)/(ZP(J+1)-ZL(J))	CONS 480
	ZATIOH(JG)=ZP(J+1)-ZL(J)	CONS 490
150	CONTINUE	CONS 500
	ZATIOK(NZG)=0.	CONS 510
	ZATIOH(NZG)=0.	CONS 520
CE	*****	CONS 530
CB	PASS THROUGH ALL BLOCKS TO CALCULATE THE GAP CORRECTION FACTORS	CONS 540
	DO 230 L=1,LMAX	CONS 550
C	DEFINE THE CORNERS OF THE CURRENT BLOCK	CONS 560
	ILS=IL(L)	CONS 570
	IHS=IH(L)	CONS 580
	JLS=JL(L)	CONS 590
	JHS=JH(L)	CONS 600
C	IS IT A COOLANT OR A SOLID MATERIAL BLOCK? -'COOL.,SOLID'-	CONS 610
	IF (MB(L).LE.0) GO TO 210	CONS 620
C/	CALCULATE RADIAL GEOMETRIC GAP CORRECTION FACTOR 4	CONS 630
C	.(RATIOB, TABLE C3)	CONS 640
	IF (RDG(L).LE.0.0) GO TO 190	CONS 650
	IG=IGR(IHS)	CONS 660
	IF (ISHAPE.EQ.1) GO TO 160	CONS 670
	X=ALOG((RL(IHS)-RDG(L))/RP(IHS))/ALOG(RP(IHS+1)/RL(IHS))	CONS 680
	GO TO 170	CONS 690
160	X=(RL(IHS)-RP(IHS)-RDG(L))/(RP(IHS+1)-RL(IHS))	CONS 700
170	DO 180 J=JLS,JHS	CONS 710
	RATIOB(IG,J)=X	CONS 720
180	CONTINUE	CONS 730
CE	*****	CONS 740
C	CALCULATE AXIAL GEOMETRIC GAP CORRECTION FACTOR 4	CONS 750
C	.(ZATIOB, TABLE C3)	CONS 760
190	IF (ZDG(L).LE.0.0) GO TO 230	CONS 770
	JG=JGZ(JHS)	CONS 780
	X=(ZL(JHS)-ZP(JHS)-ZDG(L))/(ZP(JHS+1)-ZL(JHS))	CONS 790
	DO 200 I=ILS,IHS	CONS 800
	ZATIOB(JG,I)=X	CONS 810
200	CONTINUE	CONS 820
	GO TO 230	CONS 830
C/	CALCULATE THE RADIAL CORRECTION FOR RADIATION ACROSS A COOLANT	CONS 840
210	IF (ILS.EQ.1.OR.IHS.EQ.IMAX) GO TO 212	CONS 850
	IF (ISHAPE.EQ.1) GO TO 214	CONS 860
	X=RL(ILS-1)/RL(IHS)	CONS 870
	GO TO 216	CONS 880
212	X=0.	CONS 890
	GO TO 216	CONS 900
214	X=1.0	CONS 910

216	IG=IGR(IHS)	CONS 920
	DO 220 J=JLS,JHS	CONS 930
	RATIOB(IG,J)=X	CONS 940
220	CONTINUE	CONS 950
CE	*****	CONS 960
C	HAVE ALL BLOCKS BEEN CHECKED? -'YES,NO'-	CONS 970
230	CONTINUE	CONS 980
	RETURN	CONS 990
	END	CONS1000

```

SUBROUTINE GEOMET                                GEOM 10
  INCLUDE COMDIM                                GEOM 20
C= =====GEOM 30
CB CALCULATE THE GEOMETRY FACTOR (HEAT FLOW AREA/DISTANCE) AND NODE GEOM 40
C -VOLUMES                                     GEOM 50
C= =====GEOM 60
C ERROR STOPS                                  GEOM 70
C GEOMET1 SOME POINT HAS A NEGATIVE CALCULATED VOLUME. GEOM 80
C =====GEOM 90
C= =====GEOM 100
C LOGICAL IGAP ,JGAP                          GEOM 110
C= =====GEOM 120
PI=3.1415927                                     GEOM 130
TWOPI=2.*PI                                     GEOM 140
IP=ISHAPE+1                                     GEOM 150
C PASS THROUGH ALL GRID LINES IN THE RADIAL-X DIRECTION GEOM 160
DO 200 I=1,IM                                     GEOM 170
C IS A RADIAL-X GAP ADJACENT? -'NO,YES'-        GEOM 180
IF (IGR(I).EQ.0) GO TO 100                        GEOM 190
C= =====GEOM 200
C ASSIGN THE RADIAL-X GAP INDEX                  GEOM 210
IG=IGR(I)                                         GEOM 220
IGAPE=TRUE.                                       GEOM 230
GO TO 110                                         GEOM 240
C= =====GEOM 250
C SET THE RADIAL-X GAP AVERAGES TO ZERO          GEOM 260
100 IGAPE=FALSE.                                  GEOM 270
DGR=0.0                                           GEOM 280
DGR=0.0                                           GEOM 290
C= =====GEOM 300
C PASS THROUGH ALL GRID LINES IN THE AXIAL-Y-THETA DIRECTION GEOM 310
110 DO 100 J=1,JM                                     GEOM 320
C IS AN AXIAL-Y-THETA GAP ADJACENT? -'NO,YES'-   GEOM 330
IF (JGZ(J).EQ.0) GO TO 120                        GEOM 340
C= =====GEOM 350
C ASSIGN THE AXIAL-Y-THETA GAP INDEX              GEOM 360
JG=JGZ(J)                                         GEOM 370
JGAPE=TRUE.                                       GEOM 380
GO TO 130                                         GEOM 390
C= =====GEOM 400
C SET THE AXIAL-Y-THETA GAP AVERAGES TO ZERO      GEOM 410
120 JGAPE=FALSE.                                  GEOM 420
DGIK=0.0                                           GEOM 430
DGIK=0.0                                           GEOM 440
C= =====GEOM 450

```

```

C      IS A RADIAL-X GAP ADJACENT? -'NO,YES'-                      GEOM 460
130 IF (.NOT.IGAP) GO TO 140                                         GEOM 470
C= =====GEOM 480
C      ASSIGN THE RADIAL-X GAP AVERAGES                             GEOM 490
      DGJK=(GAPR(IG,J)+GAPR(IG,J+1))*0.5                             GEOM 500
      DGR=GAPR(IG,J)                                                 GEOM 510
C= =====GEOM 520
C      IS A AXIAL-Y-THETA GAP ADJACENT? -'NO,YES'-                 GEOM 530
140 IF (.NOT.JGAP) GO TO 150                                         GEOM 540
C= =====GEOM 550
C      ASSIGN THE AXIAL-Y-THETA GAP AVERAGES                         GEOM 560
      DGJK=(GAPZ(JG,I)+GAPZ(JG,I+1))*0.5                             GEOM 570
      DGZ=GAPZ(JG,I)                                                 GEOM 580
C= =====GEOM 590
C= =====GEOM 600
150 IF (J.LE.1) GO TO 152                                           GEOM 610
      ZLX=ZL(J-1)                                                    GEOM 620
      GO TO 154                                                       GEOM 630
152 ZLX=0.0                                                         GEOM 640
154 IF (I.LE.1) GO TO 156                                           GEOM 650
      RLX=RL(I-1)                                                    GEOM 660
      GO TO 158                                                       GEOM 670
156 RLX=0.0                                                         GEOM 680
C= =====GEOM 690
C      IS THE GEOMETRY CYLINDRICAL, RECTANGULAR, OR CIRCULAR? CYLIN,RECTGEOM 700
C      -.,CIRC.,'                                                  GEOM 710
158 GO TO (100,170,180), IP                                         GEOM 720
C= =====GEOM 730
C      CALCULATE THE GEOMETRY FACTOR AND NOSE VOLUME FOR A         GEOM 740
C      CYLINDRICAL GEOMETRY                                         GEOM 750
160 RR(1,J)=(ZL(J)-ZLX-DGJK)*TWOPI/ALOG(RP(I+1)/RP(I))             GEOM 760
      RZ(1,J)=((RL(I)-DGJK)**2-RLX**2)*PI/(ZP(J+1)-ZP(J))           GEOM 770
      V(1,J)=((RL(I)-DGR)**2-RLX**2)*(ZL(J)-ZLX-DGZ)*PI            GEOM 780
      GO TO 185                                                       GEOM 790
C      CALCULATE THE GEOMETRY FACTOR AND NOSE VOLUME FOR A         GEOM 800
C      RECTANGULAR GEOMETRY                                         GEOM 810
170 RR(1,J)=HEIGHT*(ZL(J)-ZLX-DGJK)/DELX,I)                       GEOM 820
      RZ(1,J)=HEIGHT*(RL(I)-RLX-DGJK)/DELZ,J)                       GEOM 830
      V(1,J)=HEIGHT*(RL(I)-RLX-DGR)*(ZL(J)-ZLX-DGZ)                 GEOM 840
      GO TO 185                                                       GEOM 850
C      CALCULATE THE GEOMETRY FACTOR AND NOSE VOLUME FOR A         GEOM 860
C      CIRCULAR GEOMETRY                                         GEOM 870
180 RZ(1,J)=(RL(I)-RLX-DGJK)/(RP(I) *(ZP(J+1)-ZP(J))*HEIGHT)       GEOM 880
      RR(1,J)=(ZL(J)-ZLX-DGJK)/ALOG(RP(I+1)/RP(I))*HEIGHT           GEOM 890
      V(1,J)=((RL(I)-DGR)**2-RLX**2)*(ZL(J)-ZLX-DGZ)*.5*HEIGHT     GEOM 900
185 IF (V(1,J).LT.0.0) CALL ERROR (7HGEOM=T1)                      GEOM 910

```

```

CE *****GEOM 920
190 CONTINUE                                         GEOM 930
200 CONTINUE                                         GEOM 940
C=====GEOM 950
C  CLEAR-UP CALCULATED ARRAYS SO THAT ONLY PHYSICALLY MEANINGFUL  GEOM 960
C  DATA IS PRESENTED                                           GEOM 970
C  DO 210 I=1,IMAX                                             GEOM 980
C  RR(1,1)=0.0                                                 GEOM 990
C  RR(1,IMAX)=0.0                                             GEOM1000
C  RZ(1,IMAX)=0.0                                             GEOM1010
C  V(1,1)=0.0                                                 GEOM1020
210 V(1,IMAX)=0.0                                             GEOM1030
C  DO 220 J=1,JMAX                                             GEOM1040
C  V(1,J)=0.0                                                 GEOM1050
C  V(IMAX,J)=0.0                                             GEOM1060
C  RR(IMAX,J)=0.0                                             GEOM1070
C  RZ(1,J)=0.0                                               GEOM1080
220 RZ(IMAX,J)=0.0                                           GEOM1090
C=====GEOM1100
C  SHOULD THE NODE VOLUMES AND GEOMETRY FACTORS BE PRINTED? -'NO,YES'GEOM1110
C  IF (.NOT.SW(4)) RETURN                                     GEOM1120
C=====GEOM1130
C  WRITE (6,260) (ZA(I),I=1,12)                             GEOM1140
C  WRITE (6,230)                                             GEOM1150
230 FORMAT (47X,26HNODE VOLUMES (FT**3)/)                 GEOM1160
C  PRINT THE VOLUMES                                         GEOM1170
C  CALL ARRAY (IDUM,V,2)                                     GEOM1180
C=====GEOM1190
C  WRITE (6,260) (ZA(I),I=1,12)                             GEOM1200
C  WRITE (6,240) PNAME(1),PNAME(1)                         GEOM1210
240 FORMAT (15X,40,35H GEOMETRY FACTOR (HEAT FLOW AREA/A6,48H DISTANCEGEOM1220
1CE) BETWEEN POINTS (1,J) AND (I+1,J),5X,4H(FT)/)         GEOM1230
C  PRINT THE RADIAL GEOMETRICAL CONDUCTANCES               GEOM1240
C  CALL ARRAY (IDUM,RR,2)                                   GEOM1250
C=====GEOM1260
C  WRITE (6,260) (ZA(I),I=1,12)                             GEOM1270
C  WRITE (6,250) PNAME(2),PNAME(2)                         GEOM1280
250 FORMAT (15X,40,35H GEOMETRY FACTOR (HEAT FLOW AREA/A6,48H DISTANCEGEOM1290
1CE) BETWEEN POINTS (1,J) AND (I,J+1),5X,4H(FT)/)         GEOM1300
C  PRINT THE AXIAL GEOMETRICAL CONDUCTANCES               GEOM1310
C  CALL ARRAY (IDUM,RZ,2)                                   GEOM1320
C=====GEOM1330
C  RETURN                                                    GEOM1340
C=====GEOM1350
260 FORMAT (14H,30X,12A6,/)                                GEOM1360
C=====GEOM1370
END                                                         GEOM1380

```


	SUBROUTINE ARRAY (IX,X,IFY)	ARRA 10
	INCLUDE COMDIM	ARRA 20
C=	=====	ARRA 30
C=	PRINT A TWO DIMENSIONAL INTEGER ARRAY	ARRA 40
C=	=====	ARRA 50
C=	=====	ARRA 60
	DIMENSION NUMBER(60),IX (IQ,JQ),X (IQ,JQ)	ARRA 70
	DATA NUMBER /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,	ARRA 80
1	20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,	ARRA 90
2	36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,	ARRA 100
3	52,53,54,55,56,57,58,59,60/	ARRA 110
C=	=====	ARRA 120
C	WRITE THE PROPER HEADING	ARRA 130
100	WRITE (6,100) PNAME(1),PNAME(2)	ARRA 140
160	FORMAT (1H0,42X,4HTHE ,A6,27H(I) DIRECTION IS HORIZONTAL/43X,4HTHE	ARRA 150
1	,A6,25H(J) DIRECTION IS VERTICAL)	ARRA 160
	NADD=10	ARRA 170
	IF (IFY .EQ. 1) NADD=25	ARRA 180
	NCHANG=0	ARRA 190
	IF (IFY .EQ. 3) NCHANG=18	ARRA 200
130	NUM2=0	ARRA 210
140	IF (NUM2.NE.0 .AND. JMAX.GT.NCHANG) WRITE(6,105)	ARRA 220
105	FORMAT (1H1)	ARRA 230
	NUM1=NUM2+1	ARRA 240
	NUM2=NUM2+NADD	ARRA 250
	IF (NUM2 .GT. IMAX) NUM2=IMAX	ARRA 260
	IF (IFY .EQ. 1) GO TO 205	ARRA 270
	WRITE (6,210) (NUMBER(L),L=NUM1,NUM2,	ARRA 280
210	FORMAT (1H0,111,9I12)	ARRA 290
	GO TO 215	ARRA 300
205	WRITE (6,180) (NUMBER(L),L=NUM1,NUM2,	ARRA 310
180	FORMAT (1H0,5X,25I5)	ARRA 320
215	WRITE (6,190)	ARRA 330
190	FORMAT (1H0)	ARRA 340
	DO 150 JJ=1,JMAX	ARRA 350
	J=JJ	ARRA 360
	IF (SW(21)) J=JMAX-JJ+1	ARRA 370
	GO TO (151,152,154), IFY	ARRA 380
151	WRITE (6,200) J,(IX(I,J),I=NUM1,NUM2,	ARRA 390
200	FORMAT (14,1X,I6,24I5)	ARRA 400
	GO TO 150	ARRA 410
152	WRITE (6,201) J,(X(1,J),I=NUM1,NUM2)	ARRA 420
201	FORMAT (14,1P10E12,4)	ARRA 430
	GO TO 150	ARRA 440
154	WRITE (6,202) J,(X(I,J),I=NUM1,NUM2)	ARRA 450

202	FORMAT (I4,1P10E12.5)	ARRA 460
150	CONTINUE	ARRA 470
300	IF (NUM2.LT.IMAX) GO TO 140	ARRA 480
C=	=====	ARRA 490
	RETURN	ARRA 500
	END	ARRA 510

```

SUBROUTINE UPRINT                                DPRI 10
INCLUDE COMDIM                                    DPRI 20
C= =====DPRI 30
CB PRINT THE POINT AND BOUNDARY ASSIGNMENTS      DPRI 40
C= =====DPRI 50
C= =====DPRI 60
C/ PRINT THE RADIAL-X POINT, GRIDLINE, AND BOUNDARY ASSIGNMENTS DPRI 70
WRITE (6,140) (ZA(I),I=1,12),PNAME(1),PNAME(6),PNAME(7),PNAME(6),PDPR 80
1NAME(7)                                          DPRI 90
DO 110 I=1,IMAX                                  DPRI 100
TEM=RP(I)*12.0                                    DPRI 110
WRITE (6,170) I,TEM                              DPRI 120
IF (I.EQ.IMAX) GO TO 110                          DPRI 130
TEM=RL(I)*12.0                                    DPRI 140
IF (IGR(I).GT.0) GO TO 100                         DPRI 150
WRITE (6,160) I,TEM                              DPRI 160
GO TO 110                                          DPRI 170
100 IG=IGR(I)                                     DPRI 180
WRITE (6,150) I,TEM,IG                           DPRI 190
110 CONTINUE                                     DPRI 200
CE *****DPRI 210
C/ PRINT THE AXIAL-Y-THETA POINT, GRID LINES AND BOUNDARY ASSIGNMENTSDPRI 220
WRITE (6,140) (ZA(I),I=1,12),PNAME(2),PNAME(4),PNAME(5),PNAME(4),PDPR 230
1NAME(5)                                          DPRI 240
DO 130 J=1,JMAX                                  DPRI 250
TEM=7P(J)*SCALE                                  DPRI 260
WRITE (6,170) J,TEM                              DPRI 270
IF (J.EQ.JMAX) GO TO 130                          DPRI 280
TEM=ZL(J)*SCALE                                  DPRI 290
IF (JGZ(J).GT.0) GO TO 120                        DPRI 300
WRITE (6,160) J,TEM                              DPRI 310
GO TO 130                                          DPRI 320
120 JG=JGZ(J)                                     DPRI 330
WRITE (6,150) J,TEM,JG                           DPRI 340
130 CONTINUE                                     DPRI 350
CE *****DPRI 360
RETURN                                           DPRI 370
C= =====DPRI 380
140 FORMAT (1H1,30X,12A6,///,24X,A6,23H BOUNDARY ASSIGNMENTS,///,3X,8DPRI 390
1HSEQUENCE,9X,5HPOINT,9X,9HGRID LINE,6X,14HCOOLANT OR GAP,4X,6HNUMBDPRI 400
2ER,8X,8HLOCATION,8X,8HLOCATION,7X,15HBOUNDARY NUMBER,18X,2A6,4X,2ADPRI 410
36/)                                             DPRI 420
150 FORMAT (3X,16,16X,F16.4,I16)                DPRI 430
160 FORMAT (3X,16,16X,F16.4)                    DPRI 440
170 FORMAT (3X,13,3X,F16.4)                      DPRI 450
C= =====DPRI 460
END                                              DPRI 470

```

```

SUBROUTINE INITEM                                INIT 10
INCLUDE COMDIM                                INIT 20
=====                                INIT 30
C=
CB ASSIGN THE INITIAL TEMPERATURE DISTRIBUTION    INIT 40
C=
=====                                INIT 50
C ERROR STOPS=                                INIT 60
C   INITE41 THE TEMPERATURE BLOCK'S LOWER RADIAL-X BOUNDARY DOES INIT 70
C   NOT COINCIDE WITH ANY OF THE RADIAL-X GRID BOUNDARIES INIT 80
C   INITE42 THE TEMPERATURE BLOCK'S UPPER RADIAL-X BOUNDARY DOES INIT 90
C   NOT COINCIDE WITH ANY OF THE RADIAL-X GRID BOUNDARIES INIT 100
C   ITITE43 THE TEMPERATURE BLOCK'S LOWER AXIAL-Y-THETA BOUNDARY INIT 110
C   DOES NOT COINCIDE WITH ANY OF THE AXIAL-Y-THETA INIT 120
C   GRID BOUNDARIES INIT 130
C   ITITE44 THE TEMPERATURE BLOCK'S UPPER AXIAL-Y-THETA BOUNDARY INIT 140
C   DOES NOT COINCIDE WITH ANY OF THE AXIAL-Y-THETA INIT 150
C   GRID BOUNDARIES INIT 160
C   INITE47 AN INITIAL TEMPERATURE HAS NOT BEEN ASSIGNED TO SOME INIT 170
C   INTERNAL POINT. INIT 180
C=====                                INIT 190
C=
=====                                INIT 200
C SET ALL INTERNAL TEMPERATURES TO ZERO ABSOLUTE INIT 210
DO 100 I=2,IM INIT 220
DO 100 J=2,JM INIT 230
TT(I,J)=0. INIT 240
100 CONTINUE INIT 250
C=
=====                                INIT 260
C SET ALL EXTERNAL RADIAL-X COOLANT TEMPERATURES TO ZERO FAHRENHEIT INIT 270
DO 110 J=1,JMAX INIT 280
TT(1,J)=460. INIT 290
TT(1,J)=460. INIT 300
110 CONTINUE INIT 310
C=
=====                                INIT 320
C SET ALL EXTERNAL AXIAL-Y-THETA COOLANT TEMPERATURES TO ZERO INIT 330
C FAHRENHEIT INIT 340
DO 120 I=1,IMAX INIT 350
TT(I,1)=460. INIT 360
TT(I,JMAX)=460. INIT 370
120 CONTINUE INIT 380
C=
=====                                INIT 390
C=
=====                                INIT 400
C READ AND CONVERT THE GEOMETRIC DATA OF A TEMPERATURE BLOCK INIT 410
130 READ (5,270) RMIN,RMAX,ZMIN,ZMAX,FLAG1,FLAG2 INIT 420
HAVE ALL INITIAL TEMPERATURES BEEN READ? -'YES,NO'- INIT 430
IF (RMIN+RMAX+ZMIN+ZMAX.EQ.0.0) GO TO 230 INIT 440
CALL FCARD(4.0,RMIN,RMAX,ZMIN,ZMAX,FLAG1,FLAG2,DUM,DUM) INIT 450

```

```

C= =====INIT 460
C CONVERT THE TEMPERATURE BLOCK DATA TO THE CORRECT UNITS INIT 470
  IF (RMIN.LE.0.0) RMIN=1.0E-5 INIT 480
  RMIN=RMIN/12.0 INIT 490
  IF (RMAX.LE.0.0) RMAX=1.0E-5 INIT 500
  RMAX=RMAX/12.0 INIT 510
  ZMIN=ZMIN/SCALE INIT 520
  ZMAX=ZMAX/SCALE INIT 530
C= =====INIT 540
CB ASSIGN THE UPPER AND LOWER RADIAL-X LIMITS OF THE TEMPERATURE BLOCK INIT 550
C/ FIND THE INDEX OF THE LOW RADIAL-X TEMPERATURE BLOCK BOUNDARY INIT 560
  DO 140 I=1,IM INIT 570
  IF (RMIN.LE.1.00001*RL(I).AND.RMIN.GE.0.99999*RL(I)) GO TO 150 INIT 580
140 CONTINUE INIT 590
CE *****INIT 600
C THE LOW RADIAL-X BOUNDARY LIES OUTSIDE THE SYSTEM INIT 610
  CALL ERROR (7HINITEM1) INIT 620
C= =====INIT 630
C ASSIGN THE INDEX OF THE LOW RADIAL-X TEMPERATURE BLOCK BOUNDARY INIT 640
150 IMI=I+1 INIT 650
  I=I INIT 660
C/ FIND THE INDEX OF THE HIGH RADIAL-X TEMPERATURE BLOCK BOUNDARY INIT 670
  DO 160 I=I1,IM INIT 680
  IF (RMAX.LE.1.00001*RL(I).AND.RMAX.GE.0.99999*RL(I)) GO TO 170 INIT 690
160 CONTINUE INIT 700
CE *****INIT 710
C THE HIGH RADIAL-X BOUNDARY LIES OUTSIDE THE SYSTEM INIT 720
  CALL ERROR (7HINITEM2) INIT 730
C= =====INIT 740
C ASSIGN THE INDEX OF THE HIGH RADIAL-X TEMPERATURE BLOCK BOUNDARY INIT 750
170 IMA=I INIT 760
C ADJUST THE APPROPRIATE INDEX IF THE BLOCK DESCRIBES AN EXTERNAL INIT 770
C COOLANT INIT 780
  IF (IMA.EQ.1) IMI=1 INIT 790
  IF (IMI.EQ.IMAX) IMA=IMAX INIT 800
CE *****INIT 810
CB ASSIGN THE UPPER AND LOWER AXIAL-Y-THETA LIMITS OF THE TEMPERATURE INIT 820
C -BLOCK INIT 830
C/ FIND THE INDEX OF THE LOW AXIAL-Y-THETA TEMPERATURE BLOCK BOUNDARY INIT 840
  DO 180 J=1,JM INIT 850
  IF (ZMI.LE.1.00001*ZL(J).AND.ZMIN.GE.0.99999*ZL(J)) GO TO 190 INIT 860
180 CONTINUE INIT 870
CE *****INIT 880
C THE LOW AXIAL-Y-THETA BOUNDARY LIES OUTSIDE THE SYSTEM INIT 890
  CALL ERROR (7HINITEM3) INIT 900
C= =====INIT 910

```

```

C      ASSIGN THE INDEX OF THE LOW AXIAL-Y-THETA TEMPERATURE BLOCK      INIT 121
C      -BOUNDARY                                                         INIT 13
190 JMI=J+1                                                             INIT 140
      JI=J                                                                INIT 150
C/    FIND THE INDEX OF THE HIGH AXIAL-Y-THETA TEMPERATURE BLOCK        INIT 160
C      -BOUNDARY                                                         INIT 170
      DO 200 J=JI,JM                                                    INIT 180
      IF (ZMAX.LE.1.00001*ZL(J).AND.ZMAX.GE.0.99999*ZL(J)) GO TO 210    INIT 190
200 CONTINUE                                                            INIT 200
CE    *****INIT 210
C      THE HIGH AXIAL-Y-THETA BOUNDARY LIES OUTSIDE THE SYSTEM          INIT 220
C      ASSIGN THE INDEX OF THE HIGH AXIAL-Y-THETA TEMPERATURE BLOCK      INIT 230
C      BOUNDARY                                                         INIT 240
      CALL ERROR (7HINITE4)                                             INIT 250
210 JMA=J                                                                INIT 260
C      ADJUST THE APPROPRIATE INDEX IF THE BLOCK DESCRIBES AN EXTERNAL    INIT 270
C      COOLANT                                                         INIT 280
      IF (JMA.EQ.1) JMI=1                                              INIT 290
      IF (JMI.EQ.JMAX) JMA=JMAX                                         INIT 300
C      READ AND CONVERT THE TEMPERATURE DATA OF A TEMPERATURE BLOCK      INIT 310
      READ (5,271) TEM,FLAG1,FLAG2                                     INIT 320
      CALL FCARD(1.0,TEM,FLAG1,FLAG2,DUM,DUM,DUM,DUM,DUM,DUM)          INIT 330
      TEM=TEM+460.0                                                    INIT 340
CE    =====INIT 350
C      ASSIGN THE TEMPERATURES TO ALL POINTS IN THE TEMPERATURE BLOCK      INIT 360
      DO 220 I=IMI,IMA                                                  INIT 370
      DO 220 J=JMI,JMA                                                  INIT 380
      TT(I,J)=TEM                                                       INIT 390
220 CONTINUE                                                            INIT 400
      GO TO 130                                                         INIT 410
CE    =====INIT 420
CE    =====INIT 430
C/    HAS A TEMPERATURE BEEN ASSIGNED TO EACH INTERNAL POINT? -'NO,YES'-INIT 440
230 NC=NC+1                                                             INIT 450
      CARJ(NC)=0.0                                                      INIT 460
      DO 240 I=2,IM                                                    INIT 470
      DO 240 J=2,JM                                                    INIT 480
      IF (TT(I,J).EQ.0.0) GO TO 260                                     INIT 490
240 CONTINUE                                                            INIT 500
CE    *****INIT 510
C      PUT THE INITIAL TEMPERATURES IN BOTH TEMPERATURE ARRAYS          INIT 520
      DO 250 I=1,IMAX                                                    INIT 530
      DO 250 J=1,JMAX                                                    INIT 540
      T(I,J)=TT(I,J)                                                    INIT 550
250 CONTINUE                                                            INIT 560
      RETURN                                                            INIT 570

```

```

C= =====INIT1380
C   SOME INTERNAL POINT WAS NOT ASSIGNED AN INITIAL TEMPERATURE      INIT1390
  260 CALL ERROR (7*INITEM7)                                           INIT1400
C= =====INIT1410
  270 FORMAT (4E12.4,24X,A6,A2)                                         INIT1420
  271 FORMAT (E12.4,60X,A6,A2)                                         INIT1430
C= =====INIT1440
  END                                                                    INIT1450

```

	SUBROUTINE FLOWCA	FLOW 10
	INCLUDE COMJIM	FLOW 20
C	=====	FLOW 30
C	READ AND PRINT THE COOLANT DATA CARDS	FLOW 40
C=	=====	FLOW 50
C	ERROR STOPS=	FLOW 60
C	FLOWCA1 THE REYNOLDS NUMBER LIMITS ARE NOT IN SEQUENCE	FLOW 70
C	FLOWCA2 THE FLOWRATE LIMITS ARE NOT IN SEQUENCE	FLOW 80
C	FLOWCA3 THE INLET TEMPERATURE LIMITS ARE NOT IN SEQUENCE	FLOW 90
C=	=====	FLOW 100
C=	=====	FLOW 110
	DIMENSION CDIRCT(2)	FLOW 120
	DATA CDIRCT /6HPOSITION, 6HNEGATIVE/	FLOW 130
C=	=====	FLOW 140
	WRITE (6,100) (ZA(I),I=1,12)	FLOW 150
100	FORMAT (1H1,30X,12A6,///,50X,23HCOOLANT SPECIFICATIONS)	FLOW 160
C	MAKE ALL COOLANTS NON-EXISTING	FLOW 170
	DO 110 I=1,MAXFLO	FLOW 180
	FLOW(I)=0.0	FLOW 190
110	CONTINUE	FLOW 200
C	READ THE COOLANT INFORMATION	FLOW 210
120	READ (5,300) DA,D1,D2,D3,D4,FLAG1,FLAG2	FLOW 220
	N=DA+0.1	FLOW 230
C	HAVE ALL COOLANT DATA CARDS BEEN READ? -'YES,NO' -	FLOW 240
	IF (.LE.:0) GO TO 291	FLOW 250
	CALL FCARD(5.0,DA,D1,D2,D3,D4,FLAG1,FLAG2,DUM)	FLOW 260
C	RECORD THE EXISTENCE OF THE CURRENT COOLANT	FLOW 270
	FLOW(N)=1.0	FLOW 280
	WRITE (6,130) N	FLOW 290
130	FORMAT (///,27H SPECIFICATIONS FOR COOLANT,I3,/))	FLOW 300
C	*****	FLOW 310
	IP=ABS(IPATH(N))	FLOW 320
	DIRECT=CDIRCT(1)	FLOW 330
	IF (IPATH(N).GT.0) GO TO 140	FLOW 340
	DIRECT=CDIRCT(2)	FLOW 350
140	WRITE (6,150) DIRECT,PNAME(IP)	FLOW 360
150	FORMAT (31H THE COOLANT IS FLOWING IN THE ,A6,3HVE ,A6,10H DIRECTI	FLOW 370
	10N)	FLOW 380
	IF (IP.NE.3) GO TO 170	FLOW 390
	HI=HEIGHT*12.0	FLOW 400
	WRITE (6,160) HI,PNAME(6),PNAME(7)	FLOW 410
160	FORMAT (10H HEIGHT IS,F10.4,2X,2A6)	FLOW 420
C	*****	FLOW 430
C	READ THE REMAINING SPECIFICATIONS OF THIS COOLANT	FLOW 440
170	RLIM1(N)=D1	FLOW 450


```

      RLIM2(N)=D2                                FLOW 466
      RLIM3(N)=D3                                FLOW 470
      RLIM4(N)=D4                                FLOW 480
C/ CHECK TO SEE THAT THE REYNOLDS NUMBER LIMITS ARE IN SEQUENCE FLOW 490
  IF (RLIM2(N).LT.RLIM1(N)) CALL ERROR (7HFLOWCA1) FLOW 500
  IF (RLIM3(N).LT.RLIM2(N)) CALL ERROR (7HFLOWCA1) FLOW 510
  IF (RLIM4(N).LT.RLIM3(N)) CALL ERROR (7HFLOWCA1) FLOW 520
CE ***** FLOW 530
  WRITE (6,310) D1,D2,D3,D4 FLOW 540
  READ (5,300) FLODEP(N),FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N),FLAG1, FLOW 550
1FLAG2 FLOW 560
  CALL FCARD(5.0,FLODEP(N),FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N),FLAG1 FLOW 570
1,FLAG2,0JM) FLOW 580
C/ CHECK TO SEE THAT THE FLOWRATE LIMITS ARE IN SEQUENCE FLOW 590
  IF (FLIM2(N).LT.FLIM1(N)) CALL ERROR (7HFLOWCA2) FLOW 600
  IF (FLIM3(N).LT.FLIM2(N)) CALL ERROR (7HFLOWCA2) FLOW 610
  IF (FLIM4(N).LT.FLIM3(N)) CALL ERROR (7HFLOWCA2) FLOW 620
CE ***** FLOW 630
  READ (5,300) TINDEX(N),TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N),FLAG1, FLOW 640
1FLAG2 FLOW 650
  CALL FCARD(5.0,TINDEX(N),TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N),FLAG1 FLOW 660
1,FLAG2,0JM) FLOW 670
C/ CHECK TO SEE THAT THE INLET TEMPERATURE LIMITS ARE IN SEQUENCE FLOW 680
  IF (TLIM2(N).LT.TLIM1(N)) CALL ERROR (7HFLOWCA3) FLOW 690
  IF (TLIM3(N).LT.TLIM2(N)) CALL ERROR (7HFLOWCA3) FLOW 700
  IF (TLIM4(N).LT.TLIM3(N)) CALL ERROR (7HFLOWCA3) FLOW 710
CE ***** FLOW 720
  IFLO(N)=FLODEP(N)+1.01 FLOW 730
  ITIN(N)=TINDEX(N)+1.01 FLOW 740
  I=IFLO(N) FLOW 750
CB PRINT THE FLOWRATE AND INLET TEMPERATURE DEPENDENCES FLOW 760
C DETERMINE THE DEPENDENCE OF THE FLOWRATE -NONE,TIME,OUTLET,INLET FLOW 770
  GO TO (180,190,200,210), I FLOW 780
C PRINT THAT THERE ARE NO STEPCHANGES IN FLOWRATE FLOW 790
180 WRITE (6,320) FLOW 800
  GO TO 220 FLOW 810
C PRINT THAT TIME DEFINES THE STEPCHANGES IN FLOWRATE FLOW 820
190 WRITE (6,330) FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N) FLOW 830
  GO TO 220 FLOW 840
C PRINT THAT THE OUTLET TEMPERATURE DEFINES THE STEPCHANGES IN FLOW 850
C :FLOWRATE FLOW 860
200 WRITE (6,340) FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N) FLOW 870
  GO TO 220 FLOW 880
C PRINT THAT THE INLET TEMPERATURE DEFINES THE STEPCHANGES IN FLOW 890
C :FLOWRATE FLOW 900
210 WRITE (6,350) FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N) FLOW 910

```

C/	DETERMINE THE DEPENDENCE OF THE INLET TEMPERATURE	FLOW 920
C	.-'NO,TIME,FLOW,OUTLET'-	FLOW 930
	220 I=IIN(N)	FLOW 940
	GO TO (230,240,250,260), I	FLOW 950
CE	*****	FLOW 960
C	PRINT THAT THERE ARE NO STEPCHANGES IN THE INLET TEMPERATURE	FLOW 970
	230 WRITE (0,360)	FLOW 980
	GO TO 270	FLOW 990
C	PRINT THAT TIME DEFINES THE STEPCHANGES IN THE INLET TEMPERATURE	FLOW1000
	240 WRITE (0,370) TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N)	FLOW1010
	GO TO 270	FLOW1020
C	PRINT THAT THE FLOWRATE DEFINES THE STEPCHANGES IN THE INLET	FLOW1030
C	TEMPERATURE	FLOW1040
	250 WRITE (0,380) FLIM1(N),FLIM2(N),FLIM3(N),FLIM4(N)	FLOW1050
	GO TO 270	FLOW1060
C	PRINT THAT THE OUTLET TEMPERATURE DEFINES THE STEPCHANGES IN	FLOW1070
C	THE INLET TEMPERATURE	FLOW1080
	260 WRITE (0,390) TLIM1(N),TLIM2(N),TLIM3(N),TLIM4(N)	FLOW1090
CE	*****	FLOW1100
C	DOES THE INLET OR OUTLET TEMPERATURE DEFINE THE STEPCHANGES	FLOW1110
C	IN FLOWRATE? -'NO,YES'-	FLOW1120
C/	DETERMINE THE DEPENDENCE OF THE INLET TEMPERATURE	FLOW1130
C	.-'NO,TIME,FLOW,OUTLET'-	FLOW1140
CE	*****	FLOW1150
	270 IF (FLODEP(N)-2.0) 290, 280, 280.	FLOW1160
C	CONVERT THE STEPCHANGE LIMITS TO DEGREES RANKINE	FLOW1170
	280 FLIM1(N)=FLIM1(N)+460.0	FLOW1180
	FLIM2(N)=FLIM2(N)+460.0	FLOW1190
	FLIM3(N)=FLIM3(N)+460.0	FLOW1200
C	DOES THE OUTLET TEMPERATURE DEFINE THE STEPCHANGES IN INLET	FLOW1210
C	TEMPERATURE? -'NO,YES'-	FLOW1220
	290 IF (TIMEDEP(N).NE.3.0) GO TO 120	FLOW1230
C	CONVERT THE STEPCHANGE LIMITS TO DEGREES RANKINE	FLOW1240
	TLIM1(N)=TLIM1(N)+460.0	FLOW1250
	TLIM2(N)=TLIM2(N)+460.0	FLOW1260
	TLIM3(N)=TLIM3(N)+460.0	FLOW1270
	TLIM4(N)=TLIM4(N)+460.0	FLOW1280
	GO TO 120	FLOW1290
	291 NC=NC+1	FLOW1300
	CARJ(NC)=0.0	FLOW1310
	DO 292 I=1,MAXFLO	FLOW1320
	292 NFLO(I)=0	FLOW1330
	N=0	FLOW1340
	DO 293 I=1,MAXFLO	FLOW1350
	IF (FLO(I).EQ. 0.0) GO TO 293	FLOW1360
	N=N+1	FLOW1370

```

      NFLOW(N)=1                                FLOW1380
293 CONTINUE                                    FLOW1390
      RETURN                                    FLOW1400
C= =====FLOW1410
300 FORMAT (5E12.4,12X,A6,A2)                  FLOW1420
310 FORMAT (31H THE REYNOLDS NUMBER LIMITS ARE,1P4E14.4) FLOW1430
320 FORMAT (24H NO STEP CHANGES IN FLOW)        FLOW1440
330 FORMAT (42H TIME(HR) DEFINES THE STEP CHANGES IN FLOW,1P4E14.4) FLOW1450
340 FORMAT (35H OUTLET TEMPERATURE(F) DEFINES THE STEP CHANGES IN FLOWFLOW1460
      1,1P4E14.4)                                FLOW1470
350 FORMAT (55H INLET TEMPERATURE(F) DEFINES THE STEP CHANGES IN FLOW,FLOW1480
      11P4E14.4)                                FLOW1490
360 FORMAT (37H NO STEP CHANGES IN INLET TEMPERATURE) FLOW1500
370 FORMAT (55H TIME(HR) DEFINES THE STEP CHANGES IN INLET TEMPERATUREFLOW1510
      1,1P4E14.4)                                FLOW1520
380 FORMAT (62H FLOWRATE(LB/HR) DEFINES THE STEP CHANGES IN INLET TEMPFLOW1530
      1ERATURE,1P4E14.4)                        FLOW1540
390 FORMAT (58H OUTLET TEMPERATURE(F) DEFINES THE STEP CHANGES IN INLEFLOW1550
      1T TEMPERATURE,1P4E14.4)                  FLOW1560
C= =====FLOW1570
      END                                        FLOW1580

```

	SUBROUTINE TIME	TIME 10
	INCLUDE COMDIM	TIME 20
C=	=====	TIME 30
C	READ AND PRINT THE TIME STEPS	TIME 40
C=	=====	TIME 50
C	ERROR STOPS=	TIME 60
C	TIME1 THE FIRST TIME CARD DOES NOT SPECIFY TIME UNITS.	TIME 70
C	THE PROBLEM IS NOT AFFECTED IF THE STEADY STATE	TIME 80
C	OPTION IS BEING USED.	TIME 90
C	TIME2 TOO MANY TIME HISTORY CARDS HAVE BEEN READ.	TIME 100
C	TIME3 THE INPUT CONTAINS NO TIME HISTORY CARDS AND THE	TIME 110
C	STEADY STATE OPTION IS NOT BEING USED.	TIME 120
C	=====	TIME 130
C=	=====	TIME 140
	DIMENSION PTIME (20,4)	TIME 150
	EQUIVALENCE (RCP ,PTIME)	TIME 160
C=	=====	TIME 170
C=	=====	TIME 180
	M=0	TIME 190
C	READ THE NEXT TIMESTEP	TIME 200
100	M=M+1	TIME 210
	READ (5,110) FTIME(M),DTIME(M),ST,FT,FLAG1,FLAG2	TIME 220
110	FORMAT (4E12.4,24X,A6,A2)	TIME 230
C	HAVE ALL TIMECARDS BEEN READ? -'NO,YES'-	TIME 240
	IF (DTIME(M).LE.0.0) GO TO 185	TIME 250
	CALL FCARD(4.0,FTIME(M),DTIME(M),ST,FT,FLAG1,FLAG2,DUM,DUM)	TIME 260
	ITAPE(M)=SI	TIME 270
	IUNIT=FI	TIME 280
C/	ARE THE UNITS THE SAME AS ON THE LAST CARD? -'NO,YES,WRONG'-	TIME 290
	IF (IUNIT) 120, 120, 140	TIME 300
120	IF (M-1) 130, 130, 180	TIME 310
CE	*****	TIME 320
C	THE FIRST TIMECARD HAS NO UNITS	TIME 330
130	CALL ERROR (5,TIME1)	TIME 340
C/	CONVERT THE TIMES TO THE PROPER UNITS	TIME 350
140	IF (IUNIT-2) 150,160,170	TIME 360
150	D=3600.0	TIME 370
	GO TO 180	TIME 380
160	D=60.0	TIME 390
CE	*****	TIME 400
C	CONVERT FTIME AND DTIME TO HOURS.	TIME 410
180	DTIME(M)=DTIME(M)/D	TIME 420
	FTIME(M)=FTIME(M)/D	TIME 430
C	COMPENSATE FOR THE ROUNDOFF ERROR IN DTIME.	TIME 440
170	DTIME(M)=DTIME(M)+TIMER(DTIME(M))	TIME 450

	GO TO 100	TIME 460
185	IF(.NOT.SW(10).AND.M.EQ.1) CALL ERROR(5HTIME3)	TIME 470
	NC=NC+1	TIME 480
	CARD(NC)=0.0	TIME 490
	MMAX=M-1	TIME 500
C	ARE THERE TOO MANY TIMECARDS - '10,YE\$'-	TIME 510
	IF (MMAX-20) 200, 200, 190	TIME 520
C	TOO MANY DIFFERENT TIMESTEPS ARE USED	TIME 530
190	CALL ERROR (5HTIME2)	TIME 540
200	IF(SW(10).AND.MMAX.EQ.0) GO TO 250	TIME 550
C	=====	TIME 560
C	PRINT THE TIME STEPS AND ITERATION PARAMETERS	TIME 570
	DO 210 M=1,MMAX	TIME 580
	PTIME(M,1)=FTIME(M)*60.0	TIME 590
	PTIME(M,2)=FTIME(M)*3600.0	TIME 600
	PTIME(M,3)=DTIME(M)*60.0	TIME 610
210	PTIME(M,4)=DTIME(M)*3600.0	TIME 620
	WRITE (6,220) (ZA(I),I=1,12)	TIME 630
220	FORMAT (1H1,30X,12A6,///,50X,13HTIME HISTORY,///,10X,18HEND OF TIME	TIME 640
	IE PERIOD,33X,9HTIME STEP,23X,15HPRINT FREQUENCY,7X,5HHOURS,6X,7HMIT	TIME 650
	ENUTES,6X,7HSECONDS,15X,5HHOURS,6X,7HMINUTES,6X,7HSECONDS//)	TIME 660
	WRITE (6,230) (FTIME(M),PTIME(M,1),PTIME(M,2),DTIME(M),PTIME(M,3),	TIME 670
	1PTIME(M,4),1TAPE(M),M=1,MMAX/)	TIME 680
230	FORMAT (1H,1P3E13.6,7X,3E13.6,114)	TIME 690
	IF(.NOT.SW(10)) RETURN	TIME 700
	WRITE(6,240)	TIME 710
240	FORMAT(110, 881 THE ABOVE TIME HISTORY WILL BE IGNORED BECAUSE	TIME 720
	THE SLIDING STATE OPTION IS BEING USED.)	TIME 730
250	DO 260 M=1,21	TIME 740
	DTIME(M)=0.0	TIME 750
	FTIME(M)=0.0	TIME 760
260	1TAPE(M)=0	TIME 770
	MMAX=1	TIME 780
	RETURN	TIME 790
C	*****	TIME 800
C	*****	TIME 810
C	*****	TIME 820
	FUNCTION, TIMEERR(TIM)	TIME 830
C	=====	TIME 840
C	DEFINING MAXIMUM ROUNDOFF ERROR INCURRED IN CONVERTING TIME SCALE	TIME 850
C	DATA TO HOURS.	TIME 860
C	=====	TIME 870
	REAL MAG	TIME 880
C	=====	TIME 890
	DIMENSION MAG(31)	TIME 900
	DATA(MAG(I),I=1,31) / 1.0E-15,1.0E-14,1.0E-13,1.0E-12,1.0E-11,	TIME 910

```

.          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
  TIMEERR=0. TIME 970
  IF((TIM.LT.MAG(1)).OR.(TIM.LT.0.)) RETURN TIME 980
  ISUM=-10 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
  TIMEERR= MAG(16+KX) TIME1060
C ===== TIME1070
  RETURN TIME1080
  END TIME1090

```

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

	SUBROUTINE PRETEM	PRET	10
	INCLUDE COMDIM	PRET	20
C=	=====	PRET	30
C*	ERROR STOP:	PRET	40
C*	PRETEM1 THE PROBLEM SIZE DOES NOT MATCH THE INITIAL	PRET	50
C*	TEMPERATURE DISTRIBUTION DATA	PRET	60
C=	=====	PRET	70
C=	=====	PRET	80
	EQUIVALENCE (TEMP ,W)	PRET	90
	INTEGER TT	PRET	100
	DIMENSION NUMBER(60),TEMP (1)	PRET	110
	DATA NUMBER	PRET	120
1	/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,	PRET	130
2	20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,	PRET	140
3	36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,	PRET	150
	52,53,54,55,56,57,58,59,60/	PRET	160
C=	=====	PRET	170
	WRITE (6,100) (ZA(I),I=1,12)	PRET	180
100	FORMAT (1H1,30X,12A6,///,46X,32HINPUT TEMPERATURE DISTRIBUTION)	PRET	190
C=	=====	PRET	200
	CURTI=0.0	PRET	210
	NITER=0	PRET	220
C=	=====	PRET	230
C	READ THE FIRST CARD WITH THE GRID SIZE	PRET	240
	READ (5,260) CURTI, BIMAX,BJMAX,BNITER,FLAG1,FLAG2	PRET	250
	IAMAX=BIMAX +0.1	PRET	260
	JAMAX=BJMAX +0.1	PRET	270
	NITER = BNITER +0.1	PRET	280
C	IS A PREVIOUSLY CALCULATED TEMPERATURE DISTRIBUTION PROVIDED?	PRET	290
C	.-'NO,YES'-	PRET	300
	IF (IAMAX.NE.0) GO TO 105	PRET	310
	NC=NC+1	PRET	320
	CARD(NC)=0.0	PRET	330
	GO TO 190	PRET	340
C=	=====	PRET	350
C/	IS THE GRID SIZE CORRECT? -'NO,YES'-	PRET	360
105	IF (IAMAX.NE.IAMAX) GO TO 110	PRET	370
	IF (JAMAX.EQ.JMAX) GO TO 120	PRET	380
CE	*****	PRET	390
C	THE TEMPERATURE INPUT DOES NOT MATCH THE PROBLEM	PRET	400
110	CALL ERROR (7HPRETEM1)	PRET	410
C=	=====	PRET	420
C	READ THE INITIAL TEMPERATURE DISTRIBUTION	PRET	430
120	CALL FCARD (4.0,CURTI,BIMAX,BJMAX,BNITER,FLAG1,FLAG2,DUM,DUM)	PRET	440
	NT=IMAX*JMAX	PRET	450
	CALL RCARD (TEMP,NT)	PRET	460

	NT=0	PRET 460
	DO 125 J=1,JMAX	PRET 470
	DO 125 I=1,IMAX	PRET 480
	NT=NT+1	PRET 490
	T(I,J)=TEMP(NT)	PRET 500
125	CONTINUE	PRET 510
	CALL RCARD(TI,MAXFLO)	PRET 520
	CALL RCARD(TO,MAXFLO)	PRET 530
C	READ PAST BLANK CARD	PRET 540
	READ(5,270) X	PRET 550
	NC=NC+1	PRET 560
	CARD(NC)=0.0	PRET 570
C=	=====	PRET 580
C	PREPARE ALL COOLANT DATA FOR PRINTING	PRET 590
	WRITE(6,130)	PRET 600
130	FORMAT(1H0,49X,28HCOOLANT TEMPERATURES (F)/17X,14HCOOLANT NUMBER,4X,5HINLET,5X,6HOUTLET,10X,14HCOOLANT NUMBER,4X,5HINLET,5X,6HOPRET 610	PRET 620
	2UTLET)	PRET 630
	DO 150 N=1,MAXFLO,2	PRET 640
	IF(NFLO(N).EQ.0) GO TO 180	PRET 650
	N1=NFLO(N)	PRET 660
	N2=NFLO(N+1)	PRET 670
	ITI(N1)=TI(N1)-459.5	PRET 680
	ITO(N1)=TO(N1)-459.5	PRET 690
	IF(N2.EQ.0 .OR. N.EQ.15) GO TO 140	PRET 700
	ITI(N2)=TI(N2)-459.5	PRET 710
	ITO(N2)=TO(N2)-459.5	PRET 720
	WRITE(6,160) N1,ITI(N1),ITO(N1),N2,ITI(N2),ITO(N2)	PRET 730
160	FORMAT(22X,I2,I16,I11,15X,I2,I16,I11)	PRET 740
	GO TO 150	PRET 750
140	WRITE(6,160) N1,ITI(N1),ITO(N1)	PRET 760
150	CONTINUE	PRET 770
180	IF(.NOT.SW(10)) GO TO 185	PRET 780
	CURTI=0.0	PRET 790
	NITER=0	PRET 800
185	X1=CURTI*60.0	PRET 810
	X2=CURTI*3600.0	PRET 820
	WRITE(6,250) CURTI,X1,X2,NITER	PRET 830
C=	=====	PRET 840
C	CONVERT TEMPERATURES TO INTEGRAL OUTPUT UNITS	PRET 850
190	DO 200 I=1,IMAX	PRET 860
	DO 200 J=1,JMAX	PRET 870
	TT(I,J)=T(I,J)-459.5	PRET 880
200	CONTINUE	PRET 890
C	SET THE MESH CORNERS EQUAL TO ZERO	PRET 900
	TT(1,1)=0	PRET 910

```

      TT(IMAX,1)=0                                PRET 920
      TT(1,JMAX)=0                                PRET 930
      TT(IMAX,JMAX)=0                              PRET 940
CE *****
      WRITE (6,210)                                PRET 950
210  FORMAT (1H0,47X,20HTEMPERATURES      (F))    PRET 960
      CALL ARRAY (TT,DUM,1)                        PRET 970
      RETURN                                        PRET 980
      PRET 990
C= =====PRET1000
250  FORMAT (1H0,21H THE CURRENT TIME IS,F10.4,11H HOURS = ,F10.4,13PRET1010
      1H MINUTES = ,F13.5,8H SECONDS,4X,I4,31H ITERATIONS HAVE BEEN PERPRET1020
      2FORMED,/)                                  PRET1030
260  FORMAT (E12.6,3F12.1,24X,A6,A2)              PRET1040
270  FORMAT (6E12.4)                              PRET1050
280  FORMAT (1H0,5X,25I5,/)                      PRET1060
290  FORMAT (1H0)                                  PRET1070
300  FORMAT (I4,1X,I6,24I5)                      PRET1080
C* *****PRET1090
C* *****PRET1100
C* *****PRET1110
      SUBROUTINE RCARD (TR,LENGTH)                 PRET1120
C= =====PRET1130
      DIMENSION TR ( 1),TC ( 6)                  PRET1140
C= =====PRET1150
      N=LENGTH/6                                  PRET1160
      NLEFT=LENGTH-N*6                          PRET1170
      N2=0                                         PRET1180
      DO 100 I=1,N                                PRET1190
      N1=N2+1                                     PRET1200
      N2=N2+6                                     PRET1210
      READ (5,110) (TR(J),J=N1,N2),FLAG1,FLAG2   PRET1220
      CALL FCARD (6.0,TR(N1),TR(N1+1),TR(N1+2),TR(N1+3),TR(N1+4),TR(N1+5PRET1230
      1),FLAG1,FLAG2)                            PRET1240
100  CONTINUE                                     PRET1250
      IF (NLEFT.EQ.0) GO TO 130                   PRET1260
      READ (5,110) (TC(I),I=1,6),FLAG1,FLAG2     PRET1270
      CALL FCARD(6.0,TC(1),TC(2),TC(3),TC(4),TC(5),TC(6),FLAG1,FLAG2) PRET1280
      DO 120 I=1,NLEFT                            PRET1290
      J=N2+I                                       PRET1300
      TR(J)=TC(I)                                PRET1310
120  CONTINUE                                     PRET1320
130  RETURN                                       PRET1330
110  FORMAT (6E12.4,A6,A2)                       PRET1340
      END                                         PRET1350

```

	SUBROUTINE MP2	MP2	10
	INCLUDE COMDIM	MP2	20
C=	=====	MP2	30
CB	CALCULATE THE TEMPERATURE DISTRIBUTION AS A FUNCTION OF TIME	MP2	40
C=	=====	MP2	50
C	ERROR STOPS=	MP2	60
C	MP2 1 THE CURRENT TIME OF THE PREVIOUSLY PUNCHED	MP2	70
C	TEMPERATURE DISTRIBUTION IS GREATER THAN THE ENDING	MP2	80
C	TIME OF ANY, GIVEN TIMESTEP.	MP2	90
C=	=====	MP2	100
C=	=====	MP2	110
	LOGICAL NOTEST	MP2	120
	DIMENSION IMP2 (2)	MP2	130
	DATA IMP2 /6HMP2 1, 64 /	MP2	140
C=	=====	MP2	150
C=	=====	MP2	160
	CS1=0.0	MP2	170
	CS2=0.0	MP2	180
	IM45=-10	MP2	190
	FIRST=.TRUE.	MP2	200
	JP=.FALSE.	MP2	210
	NSTORE=0	MP2	220
	DO 110 J=1,IMAX	MP2	230
	DO 110 I=1,IMAX	MP2	240
	RCP(I,J)=1.0E12	MP2	250
110	CONTINUE	MP2	260
C=	=====	MP2	270
C	THE CODE COMPLETES ONE ENTIRE SET OF CALCULATIONS WITH A TIME STEP	MP2	280
C	OF 1.0E-10 SECONDS TO INITIALIZE THE PROBLEM. THE RESULTS OF THIS	MP2	290
C	ITERATION ARE NOT PRINTED OR IS THE ITERATION COUNTED. THEN THE	MP2	300
C	USER SPECIFIED ITERATIONS ARE PERFORMED.	MP2	310
C	INSERT THE INITIALIZATION ITERATION IN THE PROPER PLACE IN THE	MP2	320
C	USER SPECIFIED ITERATIONS	MP2	330
	IF (CURTI.EQ.0.0) GO TO 130	MP2	340
	DO 120 I=1,IMAX	MP2	350
	IF (CURTI.LT.FTIME(I)) GO TO 140	MP2	360
120	CONTINUE	MP2	370
	IERROR(1)=IMP2(1)	MP2	380
	IERROR(2)=IMP2(2)	MP2	390
	CALL ERROR2	MP2	400
130	I=1	MP2	410
140	NITER=NITER+1	MP2	420
	M=I-1	MP2	430
	K=MMAX	MP2	440
150	FTIME(K+1)=FTIME(K)	MP2	450

DTIME(K+1)=DTIME(K)	MP>	460
ITAPE(K+1)=ITAPE(K)	MP>	470
K=K+1	MP>	480
IF (K.GE.1) GO TO 150	MP>	490
FTIME(I)=CURTI+1.0E-10/3600.0	MP>	500
DTIME(I)=1.0E-10/3600.0	MP>	510
ITAPE(I)=0	MP>	520
MMAX=MMAK+1	MP>	530
GO TO 165	MP>	540
C=	=====MP>	550
160 FIRST=.FALSE.	MP>	560
IF (.NOT.SW(10)) GO TO 165	MP>	570
IF (CS1.LT.1.0) GO TO 162	MP>	580
GO TO 165	MP>	590
162 CALL STEADY	MP>	600
IWNS=NIITER+18	MP>	610
C START THE NEXT TIME PERIOD	MP>	620
165 M=M+1	MP>	630
CTI=CURTI	MP>	640
IC=1	MP>	650
C HAVE ALL THE TIME PERIODS BEEN FINISHED	MP>	660
IF (M.GT.MMAK) GO TO 200	MP>	670
ENDI1=FTIME(M)	MP>	680
DTSTAN=DTIME(M)	MP>	690
NTSTAN=ITAPE(M)	MP>	700
DT=DTSTAN/2.0	MP>	710
NTA=NTSTAN	MP>	720
IF (NTSTAN.EQ.0) NTA=10000	MP>	730
C=	=====MP>	740
C CALCULATE THE TEMPERATURES DURING THE TIMESTEPS IN CURRENT TIME	MP>	750
C PERIOD	MP>	760
C HAS COMPUTER TIME ALMOST RUN OUT? -'NO,YES'-	MP>	770
170 IF (WARN(I)) 181, 200, 200	MP>	780
C HAS NUMBER OF PRINTED PAGES SPECIFIED ALMOST RUN OUT? -'NO,YES'-	MP>	790
181 CALL PAGLIN(1201,5)	MP>	800
C DEFINE THE TIME IN THE MIDDLE OF THE TIMESTEP	MP>	810
180 DATI=CURTI+DTSTAN/2.0	MP>	820
C SAVE TEMPERATURE RESULTS OF NEXT TO LAST SMOOTHING ITERATION FOR	MP>	830
C CALCULATING RESIDUALS(STEADY STATE OPTION ONLY).	MP>	840
IF (.NOT.SW(10)) GO TO 182	MP>	850
IF (NIITER.NE.IWNS) GO TO 182	MP>	860
REWIND NSTORE	MP>	870
WRITE (NSTORE) ((T(I,J),I=1,IQ),J=1,IQ)	MP>	880
C CALCULATE THE MATERIAL PROPERTIES	MP>	890
182 CALL BLOCK	MP>	900
C CALCULATE THE AVERAGE CONDUCTIVITIES BETWEEN POINTS	MP>	910

	CALL CONJUC	MP>	120
C	CALCULATE THE NEXT TEMPERATURE DISTRIBUTION	MP>	130
	NOTEST=.TRUE.	MP>	140
	CALL STEP(S1H3,NOTEST)	MP>	150
C	CALCULATE THE COOLANT TEMPERATURES	MP>	160
183	CALL COOL	MP>	170
C	CALCULATE THE BOUNDARY TEMPERATURES	MP>	180
	CALL SORT	MP>	190
C=	=====	MP>	1000
	NITER=NITER+1	MP>	1010
	CURTI=CURTI+IC*DTSTAN	MP>	1020
	NITA=NITA-1	MP>	1030
C	SHOULD THE RESULTS OF THIS TIMESTEP BE PRINTED	MP>	1040
C	IF THIS IS THE LAST TIME STEP FOR THE CURRENT TIME PERIOD, AND	MP>	1050
C	SOME PRINTING HAS BEEN REQUESTED, THE RESULTS OF THIS TIMESTEP	MP>	1060
C	WILL BE PRINTED	MP>	1070
	IF(CURTI.GE.ENDTI.AND.NITAN.NE.0) GO TO 185	MP>	1080
	IF (NITA.GT.0) GO TO 190	MP>	1090
	NITA=NITAN	MP>	1100
C	PRINT THE RESULTS OF THIS TIMESTEP	MP>	1110
185	CALL PRINT	MP>	1120
C=	=====	MP>	1130
C	HAS THIS TIME PERIOD BEEN COMPLETED	MP>	1140
190	IF (CURTI.GE.ENDTI) GO TO 160	MP>	1150
	IC=IC+1	MP>	1160
	GO TO 170	MP>	1170
CE	*****	MP>	1180
C	SET THE SWITCHES TO FORCE PRINT OF DEBUG INFORMATION AT THE	MP>	1190
C	END OF THE RUN ONLY	MP>	1200
200	IF (SW(11)) SW(5)=.TRUE.	MP>	1210
	IF (SW(12)) SW(6)=.TRUE.	MP>	1220
	IF (SW(13)) SW(7)=.TRUE.	MP>	1230
	JP=.TRUE.	MP>	1240
C	PRINT THE CURRENT TEMPERATURE DISTRIBUTION	MP>	1250
	CALL PRINT	MP>	1260
C=	=====	MP>	1270
	IF (SW(3)) END FILE OUTTAP	MP>	1280
C=	=====	MP>	1290
C	IS A PUNCHED TEMPERATURE DISTRIBUTION DESIRED? -'NO,YES'-	MP>	1300
	IF (.NOT.SW(2)) RETURN	MP>	1310
C	=====	MP>	1320
C	PUNCH THE TEMPERATURE DISTRIBUTION	MP>	1330
C	=====	MP>	1340
201	CALL PUN	MP>	1350
	RETURN	MP>	1360
	END	MP>	1370

	SUBROUTINE ERROR2	ERRO	10
	INCLUDE COMDIM	ERRO	20
C=	=====	ERRO	30
C=	=====	ERRO	40
C	RECORD THE ERROR	ERRO	50
	WRITE (0,100) IERROR(1),IERROR(2)	ERRO	60
100	FORMAT (10H ERROR AT #2A6)	ERRO	70
	CALL MERR	ERRO	80
C=	=====	ERRO	90
	END	ERRO	100

```

SUBROUTINE STEADY                                STFA 10
INCLUDE COMDIM                                  STFA 20
C ===== STFA 30
C CALCULATE THE STEADY STATE TEMPERATURES.      STFA 40
C ===== STFA 50
LOGICAL ENDSW, LREAD, WRES, NOTEST, FIR2, WRISW STFA 60
DIMENSION XT(IQ,JQ), STEADY(IQ,JQ)            STFA 70
EQUIVALENCE (XT,TT), (RCP,STEADY)              STFA 80
C ===== STFA 90
C ===== STFA 100
C SET CONSTANTS AND INDICATORS.                 STFA 110
C ===== STFA 120
DATA = -1.0E+8                                  STFA 130
TOL=1.0E-4                                       STFA 140
RIFAC=2.0                                        STFA 150
DIMAX=500.0                                     STFA 160
ITMAX=500                                       STFA 170
IPSET=10                                        STFA 180
MSTORE=2                                        STFA 190
NSTORE=3                                        STFA 200
LSTORE=4                                        STFA 210
C SET AN INDICATOR TO SUPPRESS ANY FURTHER CALCULATION OF REAL STFA 220
C CAPACITANCE TERMS.                           STFA 230
CSZ=2.0                                         STFA 240
C ===== STFA 250
C INITIALIZATIONS                               STFA 260
C ===== STFA 270
CURT=0.0                                        STFA 280
VITER=0                                        STFA 290
IDI=1                                          STFA 300
VSEIT=0                                        STFA 310
N=0                                            STFA 320
NIE=0                                          STFA 330
DTFAC=1.0                                      STFA 340
FIR2=.FALSE.                                  STFA 350
NAOV=1                                         STFA 360
LREAD=.TRUE.                                  STFA 370
ENDSW=.FALSE.                                  STFA 380
WRES=.FALSE.                                  STFA 390
C ===== STFA 400
C PREPARE TO DO A SET OF ITERATIONS.            STFA 410
C ===== STFA 420
C DETERMINE THE PARAMETERS WHICH CONTROL THE TIME STEP MAGNITUDE STFA 430
C AND RATE OF INCREASE DURING THE CURRENT ITERATION SET. STFA 440
230 IF(10T.EQ.2) GO TO 250                     STFA 450

```

	NADV=1	STFA 460
	IF(FIR2) NADV=0	STFA 470
	GO TO 270	STFA 480
250	NADV=0	STFA 490
	DTFAC=DTFAC/RIFAC	STFA 500
	FAKE=.TRUE.	STFA 510
C	IF INSTABILITIES OCCURRED DURING THE PRECEDING ITERATION, AND IF	STFA 520
C	THIS IS NOT THE FIRST ITERATION SET, RECOVER FROM DRUM THE	STFA 530
C	TEMPERATURE RESULTS OF THE NEXT TO LAST SUCCESSFUL SET.	STFA 540
	ISTORE=MSTORE	STFA 550
	IF(LEAD) ISTORE=LSTORE	STFA 560
	WRISW=.FALSE.	STFA 570
	CALL STORE(ISTORE,WRISW)	STFA 580
	GO TO 270	STFA 590
260	DTFAC=DTFAC/RIFAC	STFA 600
	NADV=0	STFA 610
C	IF INSTABILITY OCCURS DURING THE FIRST ITERATION SET, RECOVER THE	STFA 620
C	INITIAL TEMPERATURES FROM DRUM.	STFA 630
	WRISW=.FALSE.	STFA 640
	CALL STORE(MSTORE,WRISW)	STFA 650
C	=====	STFA 660
C	PERFORM A SET OF ITERATIONS.	STFA 670
C	=====	STFA 680
270	DO 1000 M=1,IPSET	STFA 690
C	DETERMINE THE TIME STEP FOR EACH ITERATION. DO NOT ALLOW THE TIME	STFA 700
C	STEP TO EXCEED DTMAX HOURS.	STFA 710
	XN=N	STFA 720
	DTSTAN=DTFAC*((XN/10.0)**3+XN+1.0)	STFA 730
	IF(DTSTAN.GT.DTMAX) DTSTAN=DTMAX	STFA 740
	DT=DTSTAN/2.0	STFA 750
C	DETERMINE THE CONDUCTANCES AND PSEUDO CAPACITANCES FOR THE	STFA 760
C	CURRENT ITERATION.	STFA 770
	CALL BLOCK	STFA 780
	CALL CONJUC	STFA 790
	CALL STACAP	STFA 800
C	IF THIS IS THE FIRST ITERATION OF A PROBLEM, SAVE THE INITIAL	STFA 810
C	TEMPERATURES ON DRUM.	STFA 820
	IF(NT.NE.0) GO TO 300	STFA 830
	WRISW=.TRUE.	STFA 840
	CALL STORE(MSTORE,WRISW)	STFA 850
C	ADVANCE THE ITERATION COUNTERS.	STFA 860
300	N=N+NADV	STFA 870
	NT=NT+1	STFA 880
C	CALCULATE TEMPERATURES FOR THE CURRENT ITERATION.	STFA 890
	IF(WARN(I).LT.0.0.AND.NT.LT.ITMAX) GO TO 400	STFA 900
	ENDSW=.TRUE.	STFA 910

	NOTEST=.TRUE.	STFA 920
	CALL STEP(3410,NOTEST)	STFA 930
	GO TO 410	STFA 940
400	NOTEST=.FALSE.	STFA 950
	CALL STEP(3220,NOTEST)	STFA 960
410	CALL COUL	STFA 970
	CALL SURT	STFA 980
	IF(ENUSW) GO TO 420	STFA 990
C	IF THIS IS THE NEXT TO LAST ITERATION OF THE SET,WRITE THE	STFA1000
C	TEMPERATURE RESULTS ON DRUM FOR USE IN CHECKING CONVERGENCE	STFA1010
C	AND STABILITY.	STFA1020
	IF (M.NE.1PSET-1) GO TO 1000	STFA1030
	REWIND NSTORE	STFA1040
	WRITE (NSTORE) ((T(I,J),I=1,10),J=1,10)	STFA1050
1000	CONTINUE	STFA1060
C	ADVANCE THE SET COUNTER.	STFA1070
	NSET=NSET+1	STFA1080
C	=====	STFA1090
C	WRITE TEMPERATURE RESULTS ON TWO DIFFERENT DRUM UNITS IN	STFA1100
C	ALTERNATING SEQUENCE AS THE SETS ARE COMPLETED.SET AN INDICATOR	STFA1110
C	TO SHOW WHICH DRUM UNIT CONTAINS THE OLDEST TEMPERATURES.DO NOT	STFA1120
C	WRITE ON DRUM UNITS IF THE TIME STEP HAS REACHED 1000 HOURS.THE	STFA1130
C	OLDEST OF THE TWO AVAILABLE SETS OF TEMPERATURES ARE USED TO	STFA1140
C	REPLACE ANY SET OF RESULTS WHICH CONTAINS EITHER NEGATIVE OR	STFA1150
C	EXTREMELY LARGE TEMPERATURES.	STFA1160
C	=====	STFA1170
	IF(LREAD) GO TO 305	STFA1180
	ISTORE=NSTORE	STFA1190
	LREAD=.TRUE.	STFA1200
	GO TO 310	STFA1210
305	ISTORE=NSTORE	STFA1220
	LREAD=.FALSE.	STFA1230
310	WRISW=.TRUE.	STFA1240
	CALL STORE(ISTORE,WRISW)	STFA1250
	GO TO 320	STFA1260
C	=====	STFA1270
C	ENTER FROM SUBROUTINE PRINT TO EVALUATE RESIDUALS FOR THE FINAL	STFA1280
C	SMOOTHING ITERATION.	STFA1290
C	=====	STFA1300
	ENTRY RESID	STFA1310
	WRISW=.TRUE.	STFA1320
C	RECOVER THE TEMPERATURES WHICH WERE STORED ON DRUM FOR USE IN	STFA1330
C	CHECKING CONVERGENCE AND STABILITY.FOR THE ENTRY THROUGH RESID,	STFA1340
C	THESE TEMPERATURES WERE STORED ON DRUM IN SUBROUTINE MP2.	STFA1350
320	REWIND NSTORE	STFA1360
	READ(NSTORE) ((XT(I,J),I=1,10),J=1,10)	STFA1370

```

C =====STFA1380
C EVALUATE THE PROXIMITY TO STEADY STATE AND DETERMINE THE CURRENT STFA1390
C VALUE OF THE STABILITY PARAMETER. STFA1400
C =====STFA1410
420 DO 1500 I=2,IM STFA1420
    DO 1500 J=2,JM STFA1430
    IF (RCP(I,J).GT.1.0E10) GO TO 1500 STFA1440
    CAP = (RCP(I,J) * (T(I,J) - XT(I,J)))/DTSTAN STFA1450
    TCEVT = (T(I,J) + XT(I,J))/2.0 STFA1460
    AA1 = TCEVT - ((T(I+1,J) + XT(I+1,J))/2.0) STFA1470
    AA2 = ((T(I-1,J) + XT(I-1,J))/2.0) - TCEVT STFA1480
    AA3 = TCEVT - ((T(I,J+1) + XT(I,J+1))/2.0) STFA1490
    AA4 = ((T(I,J-1) + XT(I,J-1))/2.0) - TCEVT STFA1500
    H1 = -KX(I,J)*KX(I,J)*AA1 STFA1510
    H2 = -KX(I-1,J)*KX(I-1,J)*AA2 STFA1520
    H3 = -KZ(I,J)*KZ(I,J)*AA3 STFA1530
    H4 = -KZ(I,J-1)*KZ(I,J-1)*AA4 STFA1540
    VCRG = H1 + H2 + H3 + H4 + W(I,J) STFA1550
    AVERG = ABS(H1) + ABS(H2) + ABS(H3) + ABS(H4) + ABS(W(I,J)) STFA1560
    ACAP = ABS(CAP) STFA1570
C CALCULATE THE CONVERGENCE PARAMETER AT EVERY MATERIAL POINT IN STFA1580
C THE PROBLEM. STFA1590
    STEADY(I,J) = ACAP/AVERG STFA1600
1500 CONTINUE STFA1610
    IF (ENDSW.OR.WRES) GO TO 360 STFA1620
C =====STFA1630
C COMPARE THE CONVERGENCE PARAMETER AT EVERY MATERIAL POINT STFA1640
C AGAINST A SPECIFIED MAXIMUM ALLOWABLE VALUE TO TEST WHETHER OR STFA1650
C NOT THE STEADY STATE SOLUTION HAS BEEN ATTAINED. STFA1660
C =====STFA1670
330 DO 200 I=2,IM STFA1680
    DO 200 J=2,JM STFA1690
    IF (STEADY(I,J).GT.1.0E10) GO TO 200 STFA1700
    IF (STEADY(I,J).GE.TOL) GO TO 210 STFA1710
200 CONTINUE STFA1720
C =====STFA1730
C PREPARE TO RETURN TO SUBROUTINE MP2 AND PERFORM THE SMOOTHING STFA1740
C ITERATIONS. STFA1750
C =====STFA1760
205 CALL STACAP STFA1770
    DTIME(2)=DTSTAN/10.0 STFA1780
    FTIME(2)=DTIME(2)*19.9 STFA1790
    ITAPE(2)=5 STFA1800
    NITER=NI STFA1810
C PRINT THE STEADY STATE TEMPERATURE RESULTS BEFORE RETURNING STFA1820
C TO MP2. STFA1830

```

```

      CALL PRINT                                     STFA1840
C      SET AN INDICATOR TO SHOW THAT THE STEADY STATE ITERATIONS HAVE STFA1850
C      BEEN COMPLETED.                             STFA1860
      CS1=2.0                                       STFA1870
C      =====STFA1880
C      RETURN TO MP2 AND PERFORM THE SMOOTHING ITERATIONS. STFA1890
C      =====STFA1900
      RETURN                                       STFA1910
C      =====STFA1920
C      SET THE INDICATOR IJT TO SHOW THE MODE OF DETERMINING THE NEXT STFA1930
C      VALUE OF THE TIME STEP AND START THE NEXT SET OF ITERATIONS. STFA1940
C      =====STFA1950
210 IJT=1                                         STFA1960
      GO TO 230                                       STFA1970
C      IF INSTABILITIES OCCURRED IN SOLVING FOR TEMPERATURES,THE RETURN STFA1980
C      FROM SUBROUTINE STEP WILL BE TO THE FOLLOWING STATEMENT. STFA1990
220 IJT=2                                         STFA2000
      IF (VSETI.NE.0) GO TO 230 STFA2010
      IJT=3                                         STFA2020
      GO TO 230                                       STFA2030
C      =====STFA2040
C      PRINT TEMPERATURES AND/OR RESIDUALS AT THE END OF THE PROBLEM. STFA2050
C      =====STFA2060
C      PREPARE THE ARRAY OF CONVERGENCE PARAMETERS AT THE END OF THE STFA2070
C      PROBLEM(I.E.RESIDUALS)FOR PRINTING. STFA2080
360 DO 380 I=1,IMAX STFA2090
      DO 380 J=1,JMAX STFA2100
      IF (STEADY(I,J).GT.1.0E10) STEADY(I,J)=0.0 STFA2110
350 CONTINUE STFA2120
C      PRINT THE LAST AVAILABLE TEMPERATURE RESULTS IF THE PROBLEM IS TO STFA2130
C      TERMINATE IN THIS SUBROUTINE. STFA2140
      IF (NRES) GO TO 435 STFA2150
      NITER=NI STFA2160
      CALL TEMPS STFA2170
C      PRINT THE RESIDUALS AT THE END OF THE PROBLEM. STFA2180
435 WRITE(6,440) STFA2190
440 FORMAT(1H0,56X,9HRESIDUALS) STFA2200
      CALL ARRAY(1DUM,STEADY,2) STFA2210
      IF (NRES) RETURN STFA2220
C      PUNCH THE LAST AVAILABLE TEMPERATURE RESULTS IF THE PROBLEM IS TO STFA2230
C      TERMINATE IN THIS SUBROUTINE. STFA2240
      CALL PUN STFA2250
      CALL MERR STFA2260
C      =====STFA2270
C      *****STFA2280
C      *****STFA2290

```

```

C *****STFA2300
C =====STFA2310
SUBROUTINE STACAP STFA2320
C =====STFA2330
C SET CAPACITANCE TERMS FOR MAXIMUM STABILITY. STFA2340
C =====STFA2350
DO 50 I=2,IM STFA2360
DO 50 J=2,JM STFA2370
IF (RCP(I,J) .GT. 1.0E10) GO TO 80 STFA2390
RCP(I,J) = RR(I-1,J)*KR(I-1,J) + RR(I,J)*KR(I,J) + RZ(I,J-1)* STFA2390
1KZ(I,J-1) + RZ(I,J)*KZ(I,J) STFA2400
80 CONTINUE STFA2410
RETURN STFA2420
C =====STFA2430
C *****STFA2440
C *****STFA2450
C *****STFA2460
C =====STFA2470
SUBROUTINE STORE(ISTORE,WRISW) STFA2480
C =====STFA2490
C WRITE ARRAYS ON DRUM OR READ ARRAYS FROM DRUM. STFA2500
C =====STFA2510
C LOGICAL WRISW STFA2520
C =====STFA2530
REWIND ISTORE STFA2540
IF(WRISW) GO TO 100 STFA2550
READ (ISTORE) ((T(I,J),I=1,IM),J=1,JM), STFA2560
. (TI(I),I=1,NQ), (TJ(I),I=1,NQ), STFA2570
. ((R3BTL(I,J),I=1,IGQ),J=1,JQ), ((R3BTH(I,J),I=1,IGQ),J=1,JQ), STFA2580
. ((Z3BTL(I,J),I=1,JGQ),J=1,IQ), ((Z3BTH(I,J),I=1,JGQ),J=1,IQ) STFA2590
RETURN STFA2600
100 WRITE(ISTORE) ((T(I,J),I=1,IM),J=1,JM), STFA2610
. (TI(I),I=1,NQ), (TJ(I),I=1,NQ), STFA2620
. ((R3BTL(I,J),I=1,IGQ),J=1,JQ), ((R3BTH(I,J),I=1,IGQ),J=1,JQ), STFA2630
. ((Z3BTL(I,J),I=1,JGQ),J=1,IQ), ((Z3BTH(I,J),I=1,JGQ),J=1,IQ) STFA2640
RETURN STFA2650
C =====STFA2660
END STFA2670

```

	SUBROUTINE BLOCK	BL0C 10
	INCLUDE COMDIM	BL0C 20
C=	=====	BL0C 30
C	EVALUATE THE MATERIAL PROPERTIES OF EACH BLOCK	BL0C 40
C=	=====	BL0C 50
C=	=====	BL0C 60
	IGMAX=IGR(IM)	BL0C 70
	JGMAX=JGZ(JM)	BL0C 80
C	TURN THE GAS CONDUCTIVITY SWITCH OFF	BL0C 90
	GASE.FALSE.	BL0C 100
C	EVALUATE THE PROPERTIES OF ALL BLOCKS	BL0C 110
	DO 110 L=1,LMAX	BL0C 120
C	CALCULATE THE POINT LIMITS OF ALL BLOCKS	BL0C 130
	IHS=IH(L)	BL0C 140
	ILS=IL(L)	BL0C 150
	JHS=JH(L)	BL0C 160
	JLS=JL(L)	BL0C 170
C	IS THIS BLOCK A COOLANT ? -'YES,NO'.	BL0C 180
	NI=NB(L)	BL0C 190
	IF(NI.LE.0)GO TO 100	BL0C 200
C	CALCULATE THE BOUNDING GAPLINES OF SOLID BLOCK L	BL0C 210
	IGL=IGR(ILS-1)	BL0C 220
	IGH=IGR(IHS)	BL0C 230
	JGL=JGZ(JLS-1)	BL0C 240
	JGH=JGZ(JHS)	BL0C 250
C	CALCULATE THE PROPERTIES OF SOLID BLOCK L	BL0C 260
	CALL MADATA	BL0C 270
	GO TO 110	BL0C 280
C	CALCULATE THE BOUNDING GAPLINES OF COOLANT BLOCK L	BL0C 290
100	IF(IHS.EQ.IMAX)GO TO 120	BL0C 300
	IF(ILS.EQ.1)GO TO 130	BL0C 310
	GO TO 140	BL0C 320
120	IGH=IGMAX	BL0C 330
	IGL=IGH	BL0C 340
	GO TO 150	BL0C 350
130	IGL=1	BL0C 360
	IGH=IGLS	BL0C 370
	GO TO 150	BL0C 380
140	IGH=IGR(IHS)	BL0C 390
	IGL=IGR(ILS-1)	BL0C 400
150	IF(JHS.EQ.JMAX)GO TO 160	BL0C 410
	IF(JLS.EQ.1)GO TO 170	BL0C 420
	GO TO 160	BL0C 430
160	JGH=JGMAX	BL0C 440
	JGL=JGHS	BL0C 450

GO TO 190	BLOC 460
170 JGLS=1	BLOC 470
JGHS=JGLS	BLOC 480
GO TO 190	BLOC 490
180 JGHS=JGL(JHS)	BLOC 500
JGLS=JGZ(JLS-1)	BLOC 510
C CALCULATE THE PROPERTIES OF COOLANT BLOCK L	BLOC 520
190 CALL FLOJAT(L)	BLOC 530
C ARE ALL BLOCKS EVALUATED -'YES,NO'-	BLOC 540
110 CONTINUE	BLOC 550
C=====	BLOC 560
C ISOLATE THE CONDUCTIVITIES OF POINTS ADJACENT COOLANT BOUNDARIES.	BLOC 570
C .STORE THEM IN THE REM-RBRT AND ZEM-ZABT LOCATIONS FOR COOLANT	BLOC 580
C .BLOCK SURFACES.	BLOC 590
C=====	BLOC 600
C CONSIDER THE RADIAL-X COOLANT BOUNDARIES.	BLOC 610
DO 220 J=1,14	BLOC 620
IG=1GR(I)	BLOC 630
C EXCLUDE THE NON-GAPLINES.	BLOC 640
IF(IG,EW,0)GO TO 220	BLOC 650
DO 210 J=2,14	BLOC 660
C EXCLUDE THE POINTS WHERE A GAP IS ADJACENT TO THE GAPLINE.	BLOC 670
IF(GAPR(IG,J).GE,0.0)GO TO 210	BLOC 680
C EXCLUDE THE POINTS WHERE TWO COOLANTc ARE ADJACENT ON THE GAPLINE.	BLOC 690
IF(GAPR(IG,J).LT.(-2.5E-10))GO TO 210	BLOC 700
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE? -'NO,YES'-	BLOC 710
IF(GAPR(IG,J).LT.(-1.5E-10))GO TO 200	BLOC 720
REML(IG,J)=CONR(I,J)	BLOC 730
RBBIL(IG,J)=CONR(I+1,J)	BLOC 740
GO TO 210	BLOC 750
200 REMH(IG,J)=CONR(I+1,J)	BLOC 760
RBBH(IG,J)=CONR(I,J)	BLOC 770
210 CONTINUE	BLOC 780
220 CONTINUE	BLOC 790
C=====	BLOC 800
C CONSIDER THE AXIAL-Y-THETA COOLANT BOUNDARIES.	BLOC 810
DO 250 J=1,14	BLOC 820
JG=JGZ(J)	BLOC 830
C EXCLUDE THE NON-GAPLINES.	BLOC 840
IF(JG,EW,0)GO TO 250	BLOC 850
DO 240 I=2,14	BLOC 860
C EXCLUDE THE POINTS WHERE A GAP IS ADJACENT TO THE GAPLINE.	BLOC 870
IF(GAPZ(JG,I).GE,0.0)GO TO 240	BLOC 880
C EXCLUDE THE POINTS WHERE TWO COOLANTc ARE ADJACENT ON THE GAPLINE.	BLOC 890
IF(GAPZ(JG,I).LT.(-2.5E-10))GO TO 240	BLOC 900
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE? -'NO,YES'-	BLOC 910

	IF(GAPZ(JG,I).LT.(-1.5E-10))GO TO 230	BLnC 920
	ZEML(JG,I)=CONZ(I,J)	BLnC 930
	ZBB(L(JG,I)=CONZ(I,J+1)	BLnC 940
	GO TO 240	BLnC 950
230	ZEMH(JG,I)=CONZ(I,J+1)	BLnC 960
	ZBB(H(JG,I)=CONZ(I,J)	BLnC 970
240	CONTINUE	BLnC 980
250	CONTINUE	BLnC 990
C	=====	BLnC1000
	RETURN	BLnC1010
C	=====	BLnC1020
	END	BLnC1030

	SUBROUTINE MADATA	MADA	10
	INCLUDE COMDIM	MADA	20
C=	=====	MADA	30
C	SUPPLY THE MATERIAL THERMAL PHYSICAL PROPERTIES	MADA	40
C=	=====	MADA	50
C=	=====	MADA	60
C=	=====	MADA	70
C=	=====	MADA	80
C=	=====	MADA	90
C=	=====	MADA	100
C=	=====	MADA	110
C=	=====	MADA	120
C=	=====	MADA	130
C=	=====	MADA	140
C=	=====	MADA	150
C=	=====	MADA	160
C=	=====	MADA	170
C=	=====	MADA	180
C=	=====	MADA	190
C=	=====	MADA	200
C=	=====	MADA	210
C	IS THE MATERIAL SOLID OR GAS IN A GAP? -'GAS,SOLID'-	MADA	220
100	IF (GAS) GO TO 1510	MADA	230
C=	=====	MADA	240
CB	CALCULATE THE SOLID MATERIAL PROPERTIES	MADA	250
C/	CALCULATE THE HEATGENERATION RATE FOR EACH POINT IN THE BLOCK	MADA	260
C=	=====	MADA	270
	HR=JATI	MADA	280
	DO 270 J=JLS,JHS	MADA	290
	FTZ=ZP(J)	MADA	300
	DO 270 I=ILS,IHS	MADA	310
	FTR=RP(I)	MADA	320
	OR=T(I,J)	MADA	330
	GO TO (110,120,130,140,150,160,170,180,190,200,210,220,230,240,250	MADA	340
	1), VI	MADA	350
110	X=HEAT1(Y)	MADA	360
	GO TO 200	MADA	370
120	X=HEAT2(Y)	MADA	380
	GO TO 200	MADA	390
130	X=HEAT3(Y)	MADA	400
	GO TO 200	MADA	410
140	X=HEAT4(Y)	MADA	420
	GO TO 200	MADA	430
150	X=HEAT5(Y)	MADA	440
	GO TO 200	MADA	450

160	X=HEAT6(Y)	MANA 460
	GO TO 260	MANA 470
170	X=HEAT7(Y)	MANA 480
	GO TO 260	MANA 490
180	X=HEAT8(Y)	MANA 500
	GO TO 260	MANA 510
190	X=HEAT9(Y)	MANA 520
	GO TO 260	MANA 530
200	X=HEAT10(Y)	MANA 540
	GO TO 260	MANA 550
210	X=HEAT11(Y)	MANA 560
	GO TO 260	MANA 570
220	X=HEAT12(Y)	MANA 580
	GO TO 260	MANA 590
230	X=HEAT13(Y)	MANA 600
	GO TO 260	MANA 610
240	X=HEAT14(Y)	MANA 620
	GO TO 260	MANA 630
250	X=HEAT15(Y)	MANA 640
260	W(I,J)=X*V(I,J)	MANA 650
270	CONTINUE	MANA 660
CE	*****	MANA 670
C/	CALCULATE THE RADIAL-X CONDUCTIVITY	MANA 680
C=	=====	MANA 690
	DO 440 I=1,LS,IHS	MANA 700
	FTI=RP(I)	MANA 710
	DO 440 J=JLS,JHS	MANA 720
	FTZ=ZP(J)	MANA 730
	DR=T(I,J)	MANA 740
	GO TO (280,290,300,310,320,330,340,350,360,370,380,390,400,410,420)	MANA 750
	1), VI	MANA 760
280	X=RCOIN1(Y)	MANA 770
	GO TO 430	MANA 780
290	X=RCOIN2(Y)	MANA 790
	GO TO 430	MANA 800
300	X=RCOIN3(Y)	MANA 810
	GO TO 430	MANA 820
310	X=RCOIN4(Y)	MANA 830
	GO TO 430	MANA 840
320	X=RCOIN5(Y)	MANA 850
	GO TO 430	MANA 860
330	X=RCOIN6(Y)	MANA 870
	GO TO 430	MANA 880
340	X=RCOIN7(Y)	MANA 890
	GO TO 430	MANA 900
350	X=RCOIN8(Y)	MANA 910

	GO TO 430	MANA 920
360	X=RCON9(Y)	MANA 930
	GO TO 430	MANA 940
370	X=RCON10(Y)	MANA 950
	GO TO 430	MANA 960
380	X=RCON11(Y)	MANA 970
	GO TO 430	MANA 980
390	X=RCON12(Y)	MANA 990
	GO TO 430	MANA1000
400	X=RCON13(Y)	MANA1010
	GO TO 430	MANA1020
410	X=RCON14(Y)	MANA1030
	GO TO 430	MANA1040
420	X=RCON15(Y)	MANA1050
430	IF (X.EQ.0.0) X=1.0E-6	MANA1060
	CONR(1,J)=1.0/X	MANA1070
440	CONTINUE	MANA1080
CE	*****	MANA1090
C/	CALCULATE THE AXIAL-Y-THETA CONDUCTIVITY	MANA1100
C=	=====	MANA1110
	DO 610 I=ILS,IHS	MANA1120
	FTZ=RP(I)	MANA1130
	DO 610 J=JLS,JHS	MANA1140
	FTZ=FP(J)	MANA1150
	DR=1/(I,J)	MANA1160
	GO TO (450,460,470,480,490,500,510,520,530,540,550,560,570,580,590)	MANA1170
	1), VI	MANA1180
450	X=ACON1(Y)	MANA1190
	GO TO 600	MANA1200
460	X=ACON2(Y)	MANA1210
	GO TO 600	MANA1220
470	X=ACON3(Y)	MANA1230
	GO TO 600	MANA1240
480	X=ACON4(Y)	MANA1250
	GO TO 600	MANA1260
490	X=ACON5(Y)	MANA1270
	GO TO 600	MANA1280
500	X=ACON6(Y)	MANA1290
	GO TO 600	MANA1300
510	X=ACON7(Y)	MANA1310
	GO TO 600	MANA1320
520	X=ACON8(Y)	MANA1330
	GO TO 600	MANA1340
530	X=ACON9(Y)	MANA1350
	GO TO 600	MANA1360
540	X=ACON10(Y)	MANA1370

GO TO 600	MAHA1380
550 X=ALCON11(Y)	MAHA1390
GO TO 600	MAHA1400
560 X=ALCON12(Y)	MAHA1410
GO TO 600	MAHA1420
570 X=ALCON13(Y)	MAHA1430
GO TO 600	MAHA1440
580 X=ALCON14(Y)	MAHA1450
GO TO 600	MAHA1460
590 X=ALCON15(Y)	MAHA1470
600 IF (X.EQ.0.0) X=1.0E-6	MAHA1480
CONZ(1,J)=1.0/X	MAHA1490
610 CONTINUE	MAHA1500
C=	MAHA1510
CE *****	MAHA1520
C/ CALCULATE THE SPECIFIC HEAT	MAHA1530
C=	MAHA1540
IF(CS2.GT.1.0) GO TO 785	MAHA1550
DO 780 I=ILS,IHS	MAHA1560
DO 780 J=JLS,JHS	MAHA1570
JR=1(1,J)	MAHA1580
GO TO (620,630,640,650,660,670,680,690,700,710,720,730,740,750,760	MAHA1590
1), VI	MAHA1600
620 X=SPEC1(Y)	MAHA1610
GO TO 770	MAHA1620
630 X=SPEC2(Y)	MAHA1630
GO TO 770	MAHA1640
640 X=SPEC3(Y)	MAHA1650
GO TO 770	MAHA1660
650 X=SPEC4(Y)	MAHA1670
GO TO 770	MAHA1680
660 X=SPEC5(Y)	MAHA1690
GO TO 770	MAHA1700
670 X=SPEC6(Y)	MAHA1710
GO TO 770	MAHA1720
680 X=SPEC7(Y)	MAHA1730
GO TO 770	MAHA1740
690 X=SPEC8(Y)	MAHA1750
GO TO 770	MAHA1760
700 X=SPEC9(Y)	MAHA1770
GO TO 770	MAHA1780
710 X=SPEC10(Y)	MAHA1790
GO TO 770	MAHA1800
720 X=SPEC11(Y)	MAHA1810
GO TO 770	MAHA1820
730 X=SPEC12(Y)	MAHA1830

GO TO 770	MAHA1840
740 X=SPEC13(Y)	MAHA1850
GO TO 770	MAHA1860
750 X=SPEC14(Y)	MAHA1870
GO TO 770	MAHA1880
760 X=SPEC15(Y)	MAHA1890
770 RCP(I,J)=X+V(I,J)	MAHA1900
IF(.NOT.SN(10)) GO TO 760	MAHA1910
IF(A.GT.9.999E5) RCP(I,J)=1.0E12	MAHA1920
780 CONTINUE	MAHA1930
CE *****	MAHA1940
C/ CALCULATE THE RADIAL-X THERMAL EMISSIVITY ON THE LOW INDEX	MAHA1950
C SIDE OF THE BLOCK	MAHA1960
C=====	MAHA1970
785 IF(IGLS.LE.0) GO TO 960	MAHA1980
DO 950 J=JLS,JMS	MAHA1990
IF (GAP(IGLS,J).EQ.0.0) GO TO 950	MAHA2000
FTZ=7P(J)	MAHA2010
JR=J+JTL(IGLS,J)	MAHA2020
GO TO (790,800,810,820,830,840,850,860,870,880,890,900,910,920,930	MAHA2030
), J1	MAHA2040
790 X=EMRL1(Y)	MAHA2050
GO TO 940	MAHA2060
800 X=EMRL2(Y)	MAHA2070
GO TO 940	MAHA2080
810 X=EMRL3(Y)	MAHA2090
GO TO 940	MAHA2100
820 X=EMRL4(Y)	MAHA2110
GO TO 940	MAHA2120
830 X=EMRL5(Y)	MAHA2130
GO TO 940	MAHA2140
840 X=EMRL6(Y)	MAHA2150
GO TO 940	MAHA2160
850 X=EMRL7(Y)	MAHA2170
GO TO 940	MAHA2180
860 X=EMRL8(Y)	MAHA2190
GO TO 940	MAHA2200
870 X=EMRL9(Y)	MAHA2210
GO TO 940	MAHA2220
880 X=EMRL10(Y)	MAHA2230
GO TO 940	MAHA2240
890 X=EMRL11(Y)	MAHA2250
GO TO 940	MAHA2260
900 X=EMRL12(Y)	MAHA2270
GO TO 940	MAHA2280
910 X=EMRL13(Y)	MAHA2290

```

      GO TO 940
920  X=EWR14(Y)
      GO TO 940
930  X=EWR15(Y)
940  IF (X.EQ.0.0) X=1.0E-6
      REML(IGLS,J)=1.0/X
950  CONTINUE
CE *****
C/  CALCULATE THE RAJIAL-X THERMAL EMISSIVITY ON THE HIGH INDEX
C    SIDE OF THE BLOCK
C=  =====
960  IF (IGHS.LE.0) GO TO 1140
      DO 1130 J=JLS,JHS
      IF (GAPR(IGHS,J).EQ.0.0) GO TO 1130
      FTZ=ZP(J)
      DR=H3TH(IGHS,J)
      DO 10 (970,980,990,1000,1010,1020,1030,1040,1050,1060,1070,1080,1090,1100,1110), NI
970  X=EWRH1(Y)
980  X=EWRH2(Y)
990  X=EWRH3(Y)
1000 X=EWRH4(Y)
1010 X=EWRH5(Y)
1020 X=EWRH6(Y)
1030 X=EWRH7(Y)
1040 X=EWRH8(Y)
1050 X=EWRH9(Y)
1060 X=EWRH10(Y)
1070 X=EWRH11(Y)
1080 X=EWRH12(Y)
1090 X=EWRH13(Y)
1100 X=EWRH14(Y)
      GO TO 1120

```

MA7A2300
 MA7A2310
 MA7A2320
 MA7A2330
 MA7A2340
 MA7A2350
 MA7A2360
 MA7A2370
 MA7A2380
 MA7A2390
 MA7A2400
 MA7A2410
 MA7A2420
 MA7A2430
 MA7A2440
 MA7A2450
 MA7A2460
 MA7A2470
 MA7A2480
 MA7A2490
 MA7A2500
 MA7A2510
 MA7A2520
 MA7A2530
 MA7A2540
 MA7A2550
 MA7A2560
 MA7A2570
 MA7A2580
 MA7A2590
 MA7A2600
 MA7A2610
 MA7A2620
 MA7A2630
 MA7A2640
 MA7A2650
 MA7A2660
 MA7A2670
 MA7A2680
 MA7A2690
 MA7A2700
 MA7A2710
 MA7A2720
 MA7A2730
 MA7A2740
 MA7A2750

1110	X=EMRH15(Y)	MANA2760
1120	IF (X.EQ.0.0) X=1.0E-6	MANA2770
	REMH(IGHS,J)=1.0/X	MANA2780
1130	CONTINUE	MANA2790
CE	*****	MANA2800
C/	CALCULATE THE AXIAL-Y-THETA THERMAL PERMISSIVITY ON THE LOW INDEX	MANA2810
C	SIDE OF THE BLOCK	MANA2820
C=	=====	MANA2830
1140	IF (JGLS.LE.0) GO TO 1320	MANA2840
	DO 1310 I=ILS,IMS	MANA2850
	IF (GAPL(JGLS,I).EQ.0.0) GO TO 1310	MANA2860
	FTR=RP(I)	MANA2870
	DR=Z4BTL(JGLS,I)	MANA2880
	GO TO (1150,1160,1170,1180,1190,1200,1210,1220,1230,1240,1250,1260	MANA2890
	1,1270,1280,1290), NI	MANA2900
1150	X=EMAL1(Y)	MANA2910
	GO TO 1300	MANA2920
1160	X=EMAL2(Y)	MANA2930
	GO TO 1300	MANA2940
1170	X=EMAL3(Y)	MANA2950
	GO TO 1300	MANA2960
1180	X=EMAL4(Y)	MANA2970
	GO TO 1300	MANA2980
1190	X=EMAL5(Y)	MANA2990
	GO TO 1300	MANA3000
1200	X=EMAL6(Y)	MANA3010
	GO TO 1300	MANA3020
1210	X=EMAL7(Y)	MANA3030
	GO TO 1300	MANA3040
1220	X=EMAL8(Y)	MANA3050
	GO TO 1300	MANA3060
1230	X=EMAL9(Y)	MANA3070
	GO TO 1300	MANA3080
1240	X=EMAL10(Y)	MANA3090
	GO TO 1300	MANA3100
1250	X=EMAL11(Y)	MANA3110
	GO TO 1300	MANA3120
1260	X=EMAL12(Y)	MANA3130
	GO TO 1300	MANA3140
1270	X=EMAL13(Y)	MANA3150
	GO TO 1300	MANA3160
1280	X=EMAL14(Y)	MANA3170
	GO TO 1300	MANA3180
1290	X=EMAL15(Y)	MANA3190
1300	IF (X.EQ.0.0) X=1.0E-6	MANA3200
	ZEML(JGLS,I)=1.0/X	MANA3210

```

1310 CONTINUE
CE *****
C/ CALCULATE THE AXIAL-Y-THETA THERMAL EMISSIVITY ON THE HIGH INDEX
C SIDE OF THE BLOCK
C= =====
1320 IF (JGHS.LE.0) GO TO 1500
    DO 1490 I=ILS,IHS
    IF (GAPZ(JGHS,I).EQ.0.0) GO TO 1490
    FTR=RP(1)
    DR=ZBZTR(JGHS,I)
    GO TO (1330,1340,1350,1360,1370,1380,1390,1400,1410,1420,1430,1440)
    1,1450,1460,1470), N1
1330 X=EMAH1(Y)
    GO TO 1480
1340 X=EMAH2(Y)
    GO TO 1480
1350 X=EMAH3(Y)
    GO TO 1480
1360 X=EMAH4(Y)
    GO TO 1480
1370 X=EMAH5(Y)
    GO TO 1480
1380 X=EMAH6(Y)
    GO TO 1480
1390 X=EMAH7(Y)
    GO TO 1480
1400 X=EMAH8(Y)
    GO TO 1480
1410 X=EMAH9(Y)
    GO TO 1480
1420 X=EMAH10(Y)
    GO TO 1480
1430 X=EMAH11(Y)
    GO TO 1480
1440 X=EMAH12(Y)
    GO TO 1480
1450 X=EMAH13(Y)
    GO TO 1480
1460 X=EMAH14(Y)
    GO TO 1480
1470 X=EMAH15(Y)
1480 IF (X.EQ.0.0) X=1.0E-6
    ZEMH(JGHS,I)=1.0/X
1490 CONTINUE
CE *****
CE *****

```

MAHA3220

MAHA3230

MAHA3240

MAHA3250

MAHA3260

MAHA3270

MAHA3280

MAHA3290

MAHA3300

MAHA3310

MAHA3320

MAHA3330

MAHA3340

MAHA3350

MAHA3360

MAHA3370

MAHA3380

MAHA3390

MAHA3400

MAHA3410

MAHA3420

MAHA3430

MAHA3440

MAHA3450

MAHA3460

MAHA3470

MAHA3480

MAHA3490

MAHA3500

MAHA3510

MAHA3520

MAHA3530

MAHA3540

MAHA3550

MAHA3560

MAHA3570

MAHA3580

MAHA3590

MAHA3600

MAHA3610

MAHA3620

MAHA3630

MAHA3640

MAHA3650

MAHA3660

MAHA3670

```

1500 RETURN
C= =====MANA3680
C/ CALCULATE THE CONDUCTIVITY OF THE GAs IN THE GAP MANA3690
C= =====MANA3700
1510 DR=(TH+TB)*0.5 MANA3710
GO TO (1520,1530,1540,1550,1560,1570,1580,1590,1600,1610,1620,1630,1640,1650,1660), NI MANA3720
1520 GK=GC0N1(X) MANA3730
GO TO 1670 MANA3740
1530 GK=GC0N2(X) MANA3750
GO TO 1670 MANA3760
1540 GK=GC0N3(X) MANA3770
GO TO 1670 MANA3780
1550 GK=GC0N4(X) MANA3790
GO TO 1670 MANA3800
1560 GK=GC0N5(X) MANA3810
GO TO 1670 MANA3820
1570 GK=GC0N6(X) MANA3830
GO TO 1670 MANA3840
1580 GK=GC0N7(X) MANA3850
GO TO 1670 MANA3860
1590 GK=GC0N8(X) MANA3870
GO TO 1670 MANA3880
1600 GK=GC0N9(X) MANA3890
GO TO 1670 MANA3900
1610 GK=GC0N10(X) MANA3910
GO TO 1670 MANA3920
1620 GK=GC0N11(X) MANA3930
GO TO 1670 MANA3940
1630 GK=GC0N12(X) MANA3950
GO TO 1670 MANA3960
1640 GK=GC0N13(X) MANA3970
GO TO 1670 MANA3980
1650 GK=GC0N14(X) MANA3990
GO TO 1670 MANA4000
1660 GK=GC0N15(X) MANA4010
1670 CONTINUE MANA4020
CE *****MANA4030
RETURN MANA4040
END MANA4060
MANA4070

```



```

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 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 GCON 1(X)
FMAT1 = 0.0
10 RETURN
END

```

```

FMAT 10
FMAT 20
FMAT 30
FMAT 40
FMAT 50
FMAT 60
FMAT 70
FMAT 80
FMAT 90
FMAT 100
FMAT 110
FMAT 120
FMAT 130
FMAT 140
FMAT 150
FMAT 160
FMAT 170
FMAT 180
FMAT 190
FMAT 200
FMAT 210
FMAT 220
FMAT 230
FMAT 240
FMAT 250
FMAT 260
FMAT 270
FMAT 280
FMAT 290

```

```

FUNCTION FMAT2 (X)
  ENTRY HEAT 2(X)
  FMAT2 = 0.0
  GO TO 10
  ENTRY RCON 2(X)
  FMAT2 = 0.0
  GO TO 10
  ENTRY ALON 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 GCON 2(X)
  FMAT2 = 0.0
10 RETURN
END

```

```

FMAT 10
FMAT 20
FMAT 30
FMAT 40
FMAT 50
FMAT 60
FMAT 70
FMAT 80
FMAT 90
FMAT 100
FMAT 110
FMAT 120
FMAT 130
FMAT 140
FMAT 150
FMAT 160
FMAT 170
FMAT 180
FMAT 190
FMAT 200
FMAT 210
FMAT 220
FMAT 230
FMAT 240
FMAT 250
FMAT 260
FMAT 270
FMAT 280
FMAT 290

```

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)                                FLOD 10
INCLUDE COMDIM                                        FLOD 20
C= =====FLOD 30
C SUPPLY THE COOLANT THERMAL PHYSICAL PROPERTIES      FLOD 40
C= =====FLOD 50
C ERROR STOPS=                                       FLOD 60
C   FLODAT1 THE INDEPENDENT VALUE LIES OUTSIDE THE FLOW RATE FLOD 70
C           FUNCTION RANGES.                         FLOD 80
C   FLODAT2 THE INDEPENDENT VALUE LIES OUTSIDE THE INLET FLOD 90
C           TEMPERATURE FUNCTION RANGES.             FLOD 100
C   FLODAT3 THE REYNOLDS NUMBER LIES OUTSIDE THE SPECIFIED RANGES FLOD 110
C= =====FLOD 120
C =====FLOD 130
C =====FLOD 140
C =====FLOD 150
DIMENSION IFLODA( 4)                                FLOD 160
DATA (IFLODA(I),I=1,4) /6HFLODAT, 1H1, 1H2, 1H3/    FLOD 170
DIMENSION STA ( 2)                                  FLOD 180
INTEGER SELECT                                       FLOD 190
C= =====FLOD 200
C= =====FLOD 210
C SUPPLY THE FLOW PROPERTIES                         FLOD 220
C= =====FLOD 230
C COOLANT NUMBER STORED AS NEGATIVE MATERIAL NUMBER FLOD 240
100 N=IABS(M3(L))                                     FLOD 250
C= =====FLOD 260
C IS THIS COOLANT IN USE? -'NO,YES'-                 FLOD 270
C IF (FLOW(N).EQ.0.0) RETURN                         FLOD 280
C= =====FLOD 290
C ASSIGN THE CURRENT TIME,INLET AND OUTLET TEMPERATURE FLOD 300
HR=JATI                                              FLOD 310
TIN=T1(N)                                           FLOD 320
TOUT=TO(N)                                           FLOD 330
C= =====FLOD 340
C/ DETERMINE THE RANGE OF THE FLOW FUNCTION TO BE USED FLOD 350
C -'LOW,ERROR,MIDDLE,HIGH'-                         FLOD 360
IP=1FLO(N)                                           FLOD 370
GO TO (160,110,120,130), IP                        FLOD 380
110 X=HR                                              FLOD 390
GO TO 140                                             FLOD 400
120 X=TOUT                                           FLOD 410
GO TO 140                                             FLOD 420
130 X=TIN                                             FLOD 430
C= =====FLOD 440
C DECIDE WHICH FLOW FUNCTION TO USE                  FLOD 450

```

```

140 IF (X.LI.FLIM1(N)) GO TO 150                                FLND 460
    IF (X.LI.FLIM2(N)) GO TO 160                                FLND 470
    IF (X.LI.FLIM3(N)) GO TO 320                                FLND 480
    IF (X.LE.FLIM4(N)) GO TO 480                                FLND 490
CE *****FLND 500
C   THE INDEPENDENT VALUE LIES OUTSIDE ALL RANGES              FLND 510
150 IERROR(1)=IFLODA(1)                                         FLND 520
    IERROR(2)=IFLODA(2)                                         FLND 530
    CALL ERRJR2                                                  FLND 540
C= =====FLND 550
C/ EVALUATE THE FLOWRATE USING THE LOW RANGE FLOW FUNCTION     FLND 560
160 GO TO (170,180,190,200,210,220,230,240,250,260,270,280,290,300,310,320,330,340,350,360,370,380,390,400,410,420,430,440,450,460,470,480,490,500,510,520,530,540,550,560,570,580,590,600,610,620,630,640,650,660,670,680,690,700,710,720,730,740,750,760,770,780,790,800,810,820,830,840,850,860,870,880,890,900,910)
1) , 4                                                           FLND 580
C= =====FLND 590
C   LOWER RANGE FLOW RATE FUNCTION                              FLND 600
170 FLOW(1)=FLO1A(X)                                           FLND 610
    GO TO 640                                                    FLND 620
180 FLOW(2)=FLO2A(X)                                           FLND 630
    GO TO 640                                                    FLND 640
190 FLOW(3)=FLO3A(X)                                           FLND 650
    GO TO 640                                                    FLND 660
200 FLOW(4)=FLO4A(X)                                           FLND 670
    GO TO 640                                                    FLND 680
210 FLOW(5)=FLO5A(X)                                           FLND 690
    GO TO 640                                                    FLND 700
220 FLOW(6)=FLO6A(X)                                           FLND 710
    GO TO 640                                                    FLND 720
230 FLOW(7)=FLO7A(X)                                           FLND 730
    GO TO 640                                                    FLND 740
240 FLOW(8)=FLO8A(X)                                           FLND 750
    GO TO 640                                                    FLND 760
250 FLOW(9)=FLO9A(X)                                           FLND 770
    GO TO 640                                                    FLND 780
260 FLOW(10)=FLO10A(X)                                          FLND 790
    GO TO 640                                                    FLND 800
270 FLOW(11)=FLO11A(X)                                          FLND 810
    GO TO 640                                                    FLND 820
280 FLOW(12)=FLO12A(X)                                          FLND 830
    GO TO 640                                                    FLND 840
290 FLOW(13)=FLO13A(X)                                          FLND 850
    GO TO 640                                                    FLND 860
300 FLOW(14)=FLO14A(X)                                          FLND 870
    GO TO 640                                                    FLND 880
310 FLOW(15)=FLO15A(X)                                          FLND 890
    GO TO 640                                                    FLND 900
CE *****FLND 910

```

```

C/  EVALUATE THE FLOWRATE USING THE MIDDLE RANGE FLOW FUNCTION          FLND 920
320 GO TO (330,340,350,360,370,380,390,400,410,420,430,440,450,460,470)FLND 930
    1), N                                FLND 940
330 FLOW(1)=FLO1B(X)                                FLND 950
    GO TO 640                                FLND 960
340 FLOW(2)=FLO2B(X)                                FLND 970
    GO TO 640                                FLND 980
350 FLOW(3)=FLO3B(X)                                FLND 990
    GO TO 640                                FLND1000
360 FLOW(4)=FLO4B(X)                                FLND1010
    GO TO 640                                FLND1020
370 FLOW(5)=FLO5B(X)                                FLND1030
    GO TO 640                                FLND1040
380 FLOW(6)=FLO6B(X)                                FLND1050
    GO TO 640                                FLND1060
390 FLOW(7)=FLO7B(X)                                FLND1070
    GO TO 640                                FLND1080
400 FLOW(8)=FLO8B(X)                                FLND1090
    GO TO 640                                FLND1100
410 FLOW(9)=FLO9B(X)                                FLND1110
    GO TO 640                                FLND1120
420 FLOW(10)=FLO10B(X)                             FLND1130
    GO TO 640                                FLND1140
430 FLOW(11)=FLO11B(X)                             FLND1150
    GO TO 640                                FLND1160
440 FLOW(12)=FLO12B(X)                             FLND1170
    GO TO 640                                FLND1180
450 FLOW(13)=FLO13B(X)                             FLND1190
    GO TO 640                                FLND1200
460 FLOW(14)=FLO14B(X)                             FLND1210
    GO TO 640                                FLND1220
470 FLOW(15)=FLO15B(X)                             FLND1230
    GO TO 640                                FLND1240
CE  *****FLND1250
C/  EVALUATE THE FLOWRATE USING THE HIGH RANGE FLOW FUNCTION          FLND1260
480 GO TO (490,500,510,520,530,540,550,560,570,580,590,600,610,620,630)FLND1270
    1), N                                FLND1280
490 FLOW(1)=FLO1C(X)                                FLND1290
    GO TO 640                                FLND1300
500 FLOW(2)=FLO2C(X)                                FLND1310
    GO TO 640                                FLND1320
510 FLOW(3)=FLO3C(X)                                FLND1330
    GO TO 640                                FLND1340
520 FLOW(4)=FLO4C(X)                                FLND1350
    GO TO 640                                FLND1360
530 FLOW(5)=FLO5C(X)                                FLND1370

```

```

      GO TO 640
540 FLOW(6)=FLOWC(X)
      GO TO 640
550 FLOW(7)=FLOWC(X)
      GO TO 640
560 FLOW(8)=FLOWC(X)
      GO TO 640
570 FLOW(9)=FLOWC(X)
      GO TO 640
580 FLOW(10)=FLOWC(X)
      GO TO 640
590 FLOW(11)=FLOWC(X)
      GO TO 640
600 FLOW(12)=FLOWC(X)
      GO TO 640
610 FLOW(13)=FLOWC(X)
      GO TO 640
620 FLOW(14)=FLOWC(X)
      GO TO 640
630 FLOW(15)=FLOWC(X)
      GO TO 640
CE *****
C      DO NOT ALLOW A ZERO FLOWRATE
      640 IF (FLOW(N).EQ.0.0) FLOW(N)=1.0E-6
      FLOW(N)
C= =====
C/ DETERMINE THE RANGE OF THE INLET TEMPERATURE FUNCTION TO BE USED
C      -'LOW,ERROR,MIDDLE,HIGH'-
      IT=ITIN(N)
      GO TO (700,650,660,670), IT
      650 X=HX
      GO TO 680
      660 X=FX
      GO TO 680
      670 X=IOUT
      GO TO 680
C= =====
      680 IF (X.LT.TLIM1(N)) GO TO 690
      IF (X.LT.TLIM2(N)) GO TO 700
      IF (X.LT.TLIM3(N)) GO TO 860
      IF (X.LT.TLIM4(N)) GO TO 1020
CE *****
C      THE INDEPENDENT VALUE LIES OUTSIDE ALL RANGES
      690 IERROR(1)=IFLODA(1)
      IERROR(2)=IFLODA(3)
      CALL ERRJ2

```

```

FL0D1380
FL0D1390
FL0D1400
FL0D1410
FL0D1420
FL0D1430
FL0D1440
FL0D1450
FL0D1460
FL0D1470
FL0D1480
FL0D1490
FL0D1500
FL0D1510
FL0D1520
FL0D1530
FL0D1540
FL0D1550
FL0D1560
FL0D1570
FL0D1580
FL0D1590
FL0D1600
FL0D1610
FL0D1620
FL0D1630
FL0D1640
FL0D1650
FL0D1660
FL0D1670
FL0D1680
FL0D1690
FL0D1700
FL0D1710
FL0D1720
FL0D1730
FL0D1740
FL0D1750
FL0D1760
FL0D1770
FL0D1780
FL0D1790
FL0D1800
FL0D1810
FL0D1820
FL0D1830

```

```

C= =====FLN01840
C/ EVALUATE THE INLET TEMPERATURE USING THE LOW RANGE FUNCTION FLN01850
700 GO TO (710,720,730,740,750,760,770,780,790,800,810,820,830,840,850)FLN01860
    1), N FLN01870
710 TI(1)=TINV1A(X) FLN01880
    GO TO 1180 FLN01890
720 TI(2)=TINV2A(X) FLN01900
    GO TO 1180 FLN01910
730 TI(3)=TINV3A(X) FLN01920
    GO TO 1180 FLN01930
740 TI(4)=TINV4A(X) FLN01940
    GO TO 1180 FLN01950
750 TI(5)=TINV5A(X) FLN01960
    GO TO 1180 FLN01970
760 TI(6)=TINV6A(X) FLN01980
    GO TO 1180 FLN01990
770 TI(7)=TINV7A(X) FLN02000
    GO TO 1180 FLN02010
780 TI(8)=TINV8A(X) FLN02020
    GO TO 1180 FLN02030
790 TI(9)=TINV9A(X) FLN02040
    GO TO 1180 FLN02050
800 TI(10)=TINV10A(X) FLN02060
    GO TO 1180 FLN02070
810 TI(11)=TINV11A(X) FLN02080
    GO TO 1180 FLN02090
820 TI(12)=TINV12A(X) FLN02100
    GO TO 1180 FLN02110
830 TI(13)=TINV13A(X) FLN02120
    GO TO 1180 FLN02130
840 TI(14)=TINV14A(X) FLN02140
    GO TO 1180 FLN02150
850 TI(15)=TINV15A(X) FLN02160
    GO TO 1180 FLN02170
CE *****FLN02180
C/ EVALUATE THE INLET TEMPERATURE USING THE MIDDLE RANGE FUNCTION FLN02190
860 GO TO (870,880,890,900,910,920,930,940,950,960,970,980,990,1000,1010)FLN02200
    110), N FLN02210
870 TI(1)=TINV1B(X) FLN02220
    GO TO 1180 FLN02230
880 TI(2)=TINV2B(X) FLN02240
    GO TO 1180 FLN02250
890 TI(3)=TINV3B(X) FLN02260
    GO TO 1180 FLN02270
900 TI(4)=TINV4B(X) FLN02280
    GO TO 1180 FLN02290

```

```

910 TI(5)=TIN5B(X)                                FLND2300
GO TO 1180                                          FLND2310
920 TI(6)=TIN6B(X)                                FLND2320
GO TO 1180                                          FLND2330
930 TI(7)=TIN7B(X)                                FLND2340
GO TO 1180                                          FLND2350
940 TI(8)=TIN8B(X)                                FLND2360
GO TO 1180                                          FLND2370
950 TI(9)=TIN9B(X)                                FLND2380
GO TO 1180                                          FLND2390
960 TI(10)=TIN10B(X)                              FLND2400
GO TO 1180                                          FLND2410
970 TI(11)=TIN11B(X)                              FLND2420
GO TO 1180                                          FLND2430
980 TI(12)=TIN12B(X)                              FLND2440
GO TO 1180                                          FLND2450
990 TI(13)=TIN13B(X)                              FLND2460
GO TO 1180                                          FLND2470
1000 TI(14)=TIN14B(X)                             FLND2480
GO TO 1180                                          FLND2490
1010 TI(15)=TIN15B(X)                             FLND2500
GO TO 1180                                          FLND2510
CE *****FLND2520
C/ EVALUATE THE INLET TEMPERATURE USING THE HIGH RANGE FUNCTION FLND2530
1020 GO TO (1030,1040,1050,1060,1070,1080,1090,1100,1110,1120,1130,1140,1150,1160,1170), N FLND2540
1030 TI(1)=TIN1C(X)                                FLND2550
GO TO 1180                                          FLND2560
1040 TI(2)=TIN2C(X)                                FLND2570
GO TO 1180                                          FLND2580
1050 TI(3)=TIN3C(X)                                FLND2590
GO TO 1180                                          FLND2600
1060 TI(4)=TIN4C(X)                                FLND2610
GO TO 1180                                          FLND2620
1070 TI(5)=TIN5C(X)                                FLND2630
GO TO 1180                                          FLND2640
1080 TI(6)=TIN6C(X)                                FLND2650
GO TO 1180                                          FLND2660
1090 TI(7)=TIN7C(X)                                FLND2670
GO TO 1180                                          FLND2680
1100 TI(8)=TIN8C(X)                                FLND2690
GO TO 1180                                          FLND2700
1110 TI(9)=TIN9C(X)                                FLND2710
GO TO 1180                                          FLND2720
1120 TI(10)=TIN10C(X)                             FLND2730
GO TO 1180                                          FLND2740
                                          FLND2750

```



```

1130 TI(11)=TIN11C(X)                                FL002760
      GO TO 1180                                        FL002770
1140 TI(12)=TIN12C(X)                                FL002780
      GO TO 1180                                        FL002790
1150 TI(13)=TIN13C(X)                                FL002800
      GO TO 1180                                        FL002810
1160 TI(14)=TIN14C(X)                                FL002820
      GO TO 1180                                        FL002830
1170 TI(15)=TIN15C(X)                                FL002840
CE *****FL002850
C   SET ALL COOLANT CONDUCTIVITIES TO ZERO          FL002860
1180 DO 1190 J=JLS,JHS                                FL002870
      DO 1190 I=ILS,IHS                                FL002880
      CONZ(I,J)=1.E8                                    FL002890
      CONR(I,J)=1.E8                                    FL002900
1190 CONTINUE                                          FL002910
      M=I45(IPATH(N))                                    FL002920
C   DETERMINE THE DIRECTION OF THE FLOW -'RADIAL-X,AXIAL-Y-THETA,NORMAFL002930
C   .L'-                                              FL002940
      GO TO (1200,1240,1280), M                        FL002950
C=====FL002960
CB   CALCULATE THE HEAT TRANSFER COEFFICIENT AND SPECIFIC HEAT FOR FL002970
C   .A NORMAL COOLANT FLOW                            FL002980
1280 DR=I(13,JLS)                                      FL002990
C   CALCULATE THE SPECIFIC HEAT AT THIS LEVEL        FL003000
      RCP(N,I)=SPEC(X)                                  FL003010
C=====FL003020
CB   CALCULATE THE HEAT TRANSFER COEFFICIENT AND SPECIFIC HEAT FOR A FL003030
C   .RADIAL-A COOLANT FLOW, OR THE HEAT TRANSFER COEFFICIENT TO THE FL003040
C   .RADIAL-X WALLS FOR A NORMAL COOLANT FLOW        FL003050
C   CALCULATE THE COOLANT PROPERTIES AT ALL LEVELS   FL003060
1200 DO 1230 I=ILS,IHS                                FL003070
C=====FL003080
      IF (M.EQ. 3) GO TO 1205                            FL003090
C   SET THE LEVEL AND LOCAL COOLANT TEMPERATURE      FL003100
      FTR=RP(I)                                          FL003110
      DR=I(13,JLS)                                      FL003120
C   CALCULATE THE SPECIFIC HEAT AT THIS LEVEL        FL003130
      RCP(N,I)=SPEC(X)                                  FL003140
C   EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS FL003150
C   WHICH EXTEND OVER THE ENTIRE AXIAL LENGTH OF THE PROBLEM. FL003160
      IF(JGHS.EQ.NZG.AND.JGLS.EQ.1) GO TO 1230          FL003170
C=====FL003180
C   CALCULATE THE HEAT TRANSFER COEFFICIENTS TO THE AXIAL-Y-THETA FL003190
C   WALLS                                              FL003200
C   IS THIS AN INTERNAL COOLANT BLOCK?                FL003210

```

```

1205 IF (JGHS .NE. JGLS) GO TO 1202                                FL003220
C   SHOULD THE HEAT TRANSFER COEFFICIENT BE CALCULATED FOR THE HIGH OR FL003230
C   .THE LOW AXIAL-Y-THETA BOUNDARY OF THE EXTERNAL COOLANT BLOCK? FL003240
      IF (JGHS .EQ. 1) GO TO 1208                                FL003250
      GO TO 1207                                                  FL003260
C=====FL003270
C   IS THE INTERNAL COOLANT BLOCK ADJACENT TO AN EXTERNAL COOLANT FL003280
C   ON EITHER OF ITS AXIAL-Y-THETA BOUNDARIES? FL003290
1202 IF (JGHS .EQ. NZG) GO TO 1207                                FL003300
      IF (JGLS .EQ. 1) GO TO 1208                                FL003310
C=====FL003320
C   IS THE COOLANT BLOCK ONLY ONE POINT WIDE IN THE AXIAL-Y-THETA FL003330
C   .DIRECTION ? -'YES,NO'- FL003340
      IF (JLS.NE.JHS) GO TO 1208                                FL003350
C   CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH FL003360
C   .AXIAL-Y-THETA BOUNDARIES OF THE COOLANT BLOCK. FL003370
      STA(1)=ZBBTH(JGLS,I) FL003380
      STA(2)=ZBBTL(JGHS,I) FL003390
      CONZ(I,JLS)=1.0/HC((STA(1)+STA(2))*5) FL003400
      GO TO 1220 FL003410
C   CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR BOTH FL003420
C   .AXIAL-Y-THETA BOUNDARIES OF THE COOLANT BLOCK. FL003430
C   HIGH AXIAL-Y-THETA BOUNDARY FL003440
1208 STA(2)=ZBBTL(JGHS,I) FL003450
      CONZ(I,JHS)=1.0/HC(STA(2)) FL003460
      IF (JGLS .EQ. 1) GO TO 1220 FL003470
C   LOW AXIAL-Y-THETA BOUNDARY FL003480
1207 STA(1)=ZBBTH(JGLS,I) FL003490
      CONZ(I,JLS)=1.0/HC(STA(1)) FL003500
1220 CONTINUE FL003510
C   HAVE THE COOLANT PROPERTIES BEEN CALCULATED AT ALL LEVELS? FL003520
C   .-'YES,NO'- FL003530
1230 CONTINUE FL003540
C=====FL003550
      IF (M.EQ.1) RETURN FL003560
C=====FL003570
CB  CALCULATE THE HEAT TRANSFER COEFFICIENT AND SPECIFIC HEAT FOR FL003580
C   .AN AXIAL-Y-THETA COOLANT FLOW, OR THE HEAT TRANSFER COEFFICIENT TO FL003590
C   .THE AXIAL-Y-THETA WALLS FOR A NORMAL COOLANT FLOW. FL003600
C   CALCULATE THE COOLANT PROPERTIES AT ALL LEVELS FL003610
1240 DO 1270 J=JLS,JHS FL003620
C=====FL003630
      IF (M .EQ. 3) GO TO 1235 FL003640
C   SET THE LEVEL AND LOCAL COOLANT TEMPERATURE FL003650
      FTZ=ZP(J) FL003660
      DR=T(1LS,J) FL003670

```

```

C      CALCULATE THE SPECIFIC HEAT AT THIS LEVEL                                FL003680
      RCPG(N,J)=SPEC(X)                                                            FL003690
C      EXCLUDE CALCULATION OF HEAT TRANSFER COEFFICIENTS FOR COOLANTS            FL003700
C      WHICH EXTEND OVER THE ENTIRE RADIAL LENGTH OF THE PROBLEM.                FL003710
      IF(IGH5.EQ.NRG.AND.IGLS.EQ.1) GO TO 1270                                    FL003720
C=====FL003730
C      CALCULATE THE HEAT TRANSFER COEFFICIENTS TO THE RADIAL-X WALLS            FL003740
C      IS THIS AN INTERNAL COOLANT BLOCK?                                         FL003750
1235 IF (IGH5.NE.IGLS) GO TO 1242                                                  FL003760
C      SHOULD THE HEAT TRANSFER COEFFICIENT BE CALCULATED FOR THE HIGH OR LOW    FL003770
C      THE LOW RADIAL-X BOUNDARY OF THE EXTERNAL COOLANT BLOCK?                 FL003780
      IF (IGH5.EQ. 1) GO TO 1248                                                  FL003790
      GO TO 1247                                                                    FL003800
C=====FL003810
C      IS THE INTERNAL COOLANT BLOCK ADJACENT TO AN EXTERNAL COOLANT            FL003820
C      ON EITHER OF ITS RADIAL-X BOUNDARIES?                                     FL003830
1242 IF (IGH5.EQ. NRG) GO TO 1247                                                  FL003840
      IF (IGLS.EQ. 1) GO TO 1248                                                  FL003850
C=====FL003860
C      IS THE COOLANT BLOCK ONLY ONE POINT WIDE IN THE RADIAL-X                FL003870
C      DIRECTION?                                                                  FL003880
      IF (IGLS.NE. IHS) GO TO 1248                                                FL003890
C      CALCULATE AN AVERAGE HEAT TRANSFER COEFFICIENT FOR BOTH RADIAL-X        FL003900
C      BOUNDARIES OF THE COOLANT BLOCK.                                          FL003910
      STA(1)=RBBTH(IGLS,J)                                                        FL003920
      STA(2)=RBBTL(IGH5,J)                                                        FL003930
      CONR(IGLS,J)=1.0/HC((STA(1)+STA(2))*0.5)                                  FL003940
      GO TO 1260                                                                    FL003950
C      CALCULATE SEPARATE HEAT TRANSFER COEFFICIENTS FOR BOTH RADIAL-X          FL003960
C      BOUNDARIES OF THE COOLANT BLOCK.                                          FL003970
C      HIGH RADIAL-X BOUNDARY                                                     FL003980
1248 STA(2)=RBBTL(IGH5,J)                                                         FL003990
      CONR(IHS,J)=1.0/HC(STA(2))                                                  FL004000
      IF (IGLS.EQ. 1) GO TO 1260                                                  FL004010
C      LOW RADIAL-X BOUNDARY                                                      FL004020
1247 STA(1)=RBBTH(IGLS,J)                                                         FL004030
      CONR(IGLS,J)=1.0/HC(STA(1))                                                FL004040
1260 CONTINUE                                                                    FL004050
C      HAVE THE COOLANT PROPERTIES BEEN CALCULATED AT ALL LEVELS?              FL004060
C      -'YES,NO'-                                                                FL004070
1270 CONTINUE                                                                    FL004080
C=====FL004090
      RETURN                                                                      FL004100
CE *****FL004110
CE *****FL004120
CE *****FL004130

```

```

      FUNCTION SPEC (X)
C/  CALCULATE THE COOLANT SPECIFIC HEAT
C=  =====
100 GO TO (110,120,130,140,150,160,170,180,190,200,210,220,230,240,250
1), 1
110 B=SPH1(X)
    GO TO 260
120 B=SPH2(X)
    GO TO 260
130 B=SPH3(X)
    GO TO 260
140 B=SPH4(X)
    GO TO 260
150 B=SPH5(X)
    GO TO 260
160 B=SPH6(X)
    GO TO 260
170 B=SPH7(X)
    GO TO 260
180 B=SPH8(X)
    GO TO 260
190 B=SPH9(X)
    GO TO 260
200 B=SPH10(X)
    GO TO 260
210 B=SPH11(X)
    GO TO 260
220 B=SPH12(X)
    GO TO 260
230 B=SPH13(X)
    GO TO 260
240 B=SPH14(X)
    GO TO 260
250 B=SPH15(X)
260 SPEC=B
    RETURN
CE  *****FL004140
CE  *****FL004150
CE  *****FL004160
      FUNCTION HC (X)
C/  CALCULATE THE HEAT TRANSFER COEFFICIENT
C=  =====
    ST=X
    GO TO (100,140,180,220,260,300,340,380,420,460,500,540,580,620,660
1), 1
C=  =====FL004170
      FL004180
      FL004190
      FL004200
      FL004210
      FL004220
      FL004230
      FL004240
      FL004250
      FL004260
      FL004270
      FL004280
      FL004290
      FL004300
      FL004310
      FL004320
      FL004330
      FL004340
      FL004350
      FL004360
      FL004370
      FL004380
      FL004390
      FL004400
      FL004410
      FL004420
      FL004430
      FL004440
      FL004450
      FL004460
      FL004470
      FL004480
      FL004490
      FL004500
      FL004510
      FL004520
      FL004530
      FL004540
      FL004550
      FL004560
      FL004570
      FL004580
      FL004590

```

100	RE=REYN1(X)	FL004600
	IA=SELEC1(N)	FL004610
	GO TO (110,120,130), IA	FL004620
110	B=H1A(X)	FL004630
	GO TO 700	FL004640
120	B=H1B(X)	FL004650
	GO TO 700	FL004660
130	B=H1C(X)	FL004670
	GO TO 700	FL004680
C=	=====	FL004690
140	RE=REYN2(X)	FL004700
	IA=SELEC1(N)	FL004710
	GO TO (150,160,170), IA	FL004720
150	B=H2A(X)	FL004730
	GO TO 700	FL004740
160	B=H2B(X)	FL004750
	GO TO 700	FL004760
170	B=H2C(X)	FL004770
	GO TO 700	FL004780
C=	=====	FL004790
180	RE=REYN3(X)	FL004800
	IA=SELEC1(N)	FL004810
	GO TO (190,200,210), IA	FL004820
190	B=H3A(X)	FL004830
	GO TO 700	FL004840
200	B=H3B(X)	FL004850
	GO TO 700	FL004860
210	B=H3C(X)	FL004870
	GO TO 700	FL004880
C=	=====	FL004890
220	RE=REYN4(X)	FL004900
	IA=SELEC1(N)	FL004910
	GO TO (230,240,250), IA	FL004920
230	B=H4A(X)	FL004930
	GO TO 700	FL004940
240	B=H4B(X)	FL004950
	GO TO 700	FL004960
250	B=H4C(X)	FL004970
	GO TO 700	FL004980
C=	=====	FL004990
260	RE=REYN5(X)	FL005000
	IA=SELEC1(N)	FL005010
	GO TO (270,280,290), IA	FL005020
270	B=H5A(X)	FL005030
	GO TO 700	FL005040
280	B=H5B(X)	FL005050

	GO TO 700	FL005060
290	B=H0C(X)	FL005070
	GO TO 700	FL005080
C=	=====	FL005090
300	RE=REYN6(X)	FL005100
	IA=SELECT(N)	FL005110
	GO TO (310,320,330), IA	FL005120
310	B=H0A(X)	FL005130
	GO TO 700	FL005140
320	B=H0B(X)	FL005150
	GO TO 700	FL005160
330	B=H0C(X)	FL005170
	GO TO 700	FL005180
C=	=====	FL005190
340	RE=REYN7(X)	FL005200
	IA=SELECT(N)	FL005210
	GO TO (350,360,370), IA	FL005220
350	B=H7A(X)	FL005230
	GO TO 700	FL005240
360	B=H7B(X)	FL005250
	GO TO 700	FL005260
370	B=H7C(X)	FL005270
	GO TO 700	FL005280
C=	=====	FL005290
380	RE=REYN8(X)	FL005300
	IA=SELECT(N)	FL005310
	GO TO (390,400,410), IA	FL005320
390	B=H8A(X)	FL005330
	GO TO 700	FL005340
400	B=H8B(X)	FL005350
	GO TO 700	FL005360
410	B=H8C(X)	FL005370
	GO TO 700	FL005380
C=	=====	FL005390
420	RE=REYN9(X)	FL005400
	IA=SELECT(N)	FL005410
	GO TO (430,440,450), IA	FL005420
430	B=H9A(X)	FL005430
	GO TO 700	FL005440
440	B=H9B(X)	FL005450
	GO TO 700	FL005460
450	B=H9C(X)	FL005470
	GO TO 700	FL005480
C=	=====	FL005490
460	RE=REYN10(X)	FL005500
	IA=SELECT(N)	FL005510

```

      GO TO (470,480,490), IA
470  B=H10A(X)
      GO TO 700
480  B=H10B(X)
      GO TO 700
490  B=H10C(X)
      GO TO 700
C=
=====
500  RE=REYN11(X)
      IA=SELECT(N)
      GO TO (510,520,530), IA
510  B=H11A(X)
      GO TO 700
520  B=H11B(X)
      GO TO 700
530  B=H11C(X)
      GO TO 700
C=
=====
540  RE=REYN12(X)
      IA=SELECT(N)
      GO TO (550,560,570), IA
550  B=H12A(X)
      GO TO 700
560  B=H12B(X)
      GO TO 700
570  B=H12C(X)
      GO TO 700
C=
=====
580  RE=REYN13(X)
      IA=SELECT(N)
      GO TO (590,600,610), IA
590  B=H13A(X)
      GO TO 700
600  B=H13B(X)
      GO TO 700
610  B=H13C(X)
      GO TO 700
C=
=====
620  RE=REYN14(X)
      IA=SELECT(N)
      GO TO (630,640,650), IA
630  B=H14A(X)
      GO TO 700
640  B=H14B(X)
      GO TO 700
650  B=H14C(X)

```

FL005520
 FL005530
 FL005540
 FL005550
 FL005560
 FL005570
 FL005580
 FL005590
 FL005600
 FL005610
 FL005620
 FL005630
 FL005640
 FL005650
 FL005660
 FL005670
 FL005680
 FL005690
 FL005700
 FL005710
 FL005720
 FL005730
 FL005740
 FL005750
 FL005760
 FL005770
 FL005780
 FL005790
 FL005800
 FL005810
 FL005820
 FL005830
 FL005840
 FL005850
 FL005860
 FL005870
 FL005880
 FL005890
 FL005900
 FL005910
 FL005920
 FL005930
 FL005940
 FL005950
 FL005960
 FL005970

```

      GO TO 700
C= =====FL0D5980
060 RE=REYNIS(X)
      IA=SELECT(N)
      GO TO (670,680,690), IA
070 B=HISA(X)
      GO TO 700
080 B=HISB(X)
      GO TO 700
090 B=HISC(X)
C= =====FL0D6070
700 C=B
      IF (C.EQ.0.0) C=1.0E-6
      HC=C
      RETURN
CE *****FL0D6130
CE *****FL0D6140
CE *****FL0D6150
      INTEGER FUNCTIONSELECT(N)
C= =====FL0D6170
C   DOES THE REYNOLDS NUMBER LIE WITHIN THE SPECIFIED RANGE?
C   .-'NO,YES'-
      IF (RE.LT.RLIM1(N).OR.RE.GT.RLIM4(N)) GO TO 120
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 110
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
110 SELECT=2
      RETURN
C= =====FL0D6340
C   THE REYNOLDS NUMBER LIES OUTSIDE THE SPECIFIED RANGE
120 IERROR(1)=IFLODA(1)
      IERROR(2)=IFLODA(4)
      CALL ERKOR2
C= =====FL0D6390
      END
      FL0D6400

```



```

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

```

```

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

```

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

NOTE:

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

```

SUBROUTINE CONDOC                                COND 10
INCLUDE COMDIM                                    COND 20
C= =====COND 30
CB CALCULATE THE AVERAGE CONDUCTIVITIES BETWEEN POINTS. RADIATION COND 40
C ACROSS GAPS IS INCLUDED COND 50
C= =====COND 60
C ERROR MESSAGES COND 70
C CONDOC1 , CONDOC2 COND 80
C THESE MESSAGES INDICATE THAT RADIATION BETWEEN THE COND 90
C BOUNDARIES OF AN INTERNAL COOLANT BLOCK WAS NEGLECTEDCOND 100
C IN ORDER TO COMPLETE AN ITERATION. A SMALLER TIME COND 110
C STEP SHOULD BE USED IF THE NEGLECT OF RADIATION IS COND 120
C SIGNIFICANT. COND 130
C COND 140
C CONDOC1 RADIATION WAS NEGLECTED BETWEEN RADIAL-X GAPLINES IGLCOND 150
C AND JGH AT AXIAL-Y-THETA POINT LEVEL J. COND 160
C CONDOC2 RADIATION WAS NEGLECTED BETWEEN AXIAL-Y-THETA COND 170
C GAPLINES JGL AND JGH AT RADIAL-X POINT LEVEL I. COND 180
C =====COND 190
C= =====COND 200
LOGICAL NOMES COND 210
DIMENSION ICOND(8) COND 220
C= =====COND 230
DATA(ICOND(1),I=1,8) / 1H1,5H IGL=,5H JGH=,3H J=,1H2,5H JGL=, COND 240
. 5H JGH=,3H I= / COND 250
1000 FORMAT(16H ERROR AT CONDOC,A1,17H ON ITERATION J0.,I4,3H AT,A5,I2,COND 260
. A5,I2,A3,I2) COND 270
C= =====COND 280
C SET THE CORRECT ENTRY FOR MADATA COND 290
GASE=TRUE. COND 300
C= =====COND 310
IGMAX=IGR(IM) COND 320
JGMAX=JGZ(JM) COND 330
TDMIN=0.10 COND 340
NNITER=NNITER+1 COND 350
NOMES=.FALSE. COND 360
IF(.NOT.SW(10)) GO TO 100 COND 370
IF(US1.LT.1.0) NOMES=.TRUE. COND 380
100 IF(FIRST) NOMES=.TRUE. COND 390
C =====COND 400
CB CALCULATE THE AVERAGE RADIAL-X THERMAL CONDUCTIVITIES BETWEEN COND 410
C POINTS COND 420
C= =====COND 430
C CALCULATE THE RADIAL-X CONDUCTIVITIES ACROSS EACH RADIAL-X COND 440
C GRID LINE COND 450

```

```

DO 200 I=1,IM                                COND 460
IG=IGR(I)                                      COND 470
C= =====COND 480
C CONSIDER THE POINTS ALONG THE GRIDLINE                                COND 490
DO 190 J=2,JM                                COND 500
C DOES THE GRIDLINE HAVE ANYWHERE A GAP OR A COOLANT ADJACENT TO IT COND 510
C ? -'NO,YES'-                                COND 520
IF (IG.EQ.0) GO TO 130                        COND 530
C= =====COND 540
C IS A GAP A COOLANT ADJACENT AT THIS POINT? -'NO,YES'-                                COND 550
IF (GAPR(IG,J).EQ.0.0) GO TO 130              COND 560
C IS A GAP OR A COOLANT ADJACENT TO THE RADIAL-X GRIDLINE?-'COOL.,GA'COND 570
IF (GAPR(IG,J).LT.0.0) GO TO 150              COND 580
C= =====COND 590
CB CALCULATE THE AVERAGE CONDUCTIVITY WITH A GAP PRESENT                                COND 600
VI=MATRIG(IG,J)                              COND 610
TH=RBBTH(IG,J)                                COND 620
TJ=RBBTL(IG,J)                                COND 630
FTR=RP(I)                                     COND 640
FTZ=ZP(J)                                     COND 650
C CALCULATE THE LOCAL GAS CONDUCTIVITY.                                COND 660
CALL MAJDATA                                  COND 670
BX=REM(IG,J)+REML(IG,J)-1.0                   COND 680
CX=.1713E-8*(TH**2+TJ**2)*(T+Td)/BX           COND 690
X1=CONR(I,J)*RATIO3(IG,J)                     COND 700
X2=1./((GK/GAPR(IG,J)+CX)*RATIOH(IG,J))       COND 710
X3=CONR(I+1,J)                                 COND 720
C SAVE THE DATA REQUIRED FOR CALCULATING RADIAL-X GAP BOUNDARY COND 730
C TEMPERATURES BY STORING IT IN THE REM-RBBT LOCATIONS FOR THE COND 740
C CURRENT GAPLINE.                                COND 750
REML(IG,J)=CONR(I,J)                           COND 760
REM(IG,J)=CONR(I+1,J)                           COND 770
RBBTL(IG,J)=BX                                   COND 780
RBBTH(IG,J)=GK                                   COND 790
KR(I,J)=1./((X1+X2+X3)/RATIOK(IG,J))           COND 800
GO TO 180                                       COND 810
CE *****COND 820
CE *****COND 830
C ARE THE CONDUCTIVITIES ON BOTH SIDES OF THE GRID PLANE IDENTICAL COND 840
C ? -'NO,YES'-                                COND 850
130 IF (CONR(I,J).NE.CONR(I+1,J)) GO TO 140    COND 860
C ASSIGN ONE OF THEM AS THE AVERAGE CONDUCTIVITY COND 870
KR(I,J)=1.0/CONR(I,J)                          COND 880
GO TO 180                                       COND 890
C CALCULATE THE AVERAGE CONDUCTIVITY WHEN NO GAP OR COOLANT IS COND 900
C PRESENT                                       COND 910

```

140	KR(I,J)=1.0/(CONR(I+1,J)+(CONR(I,J)-CONR(I+1,J))*RLN(I))	COND 920
	GO TO 180	COND 930
C	CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ADJACENT	COND 940
C	IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GRIDPLANE?-'NO,YES'-	COND 950
150	IF (GAPR(IG,J).LT.(-1.5E-10)) GO TO 160	COND 960
C	CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE HIGH	COND 970
C	INDEX SIDE.	COND 980
C	EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.	COND 990
	IF(IG.EQ.IGMAX) GO TO 156	COND1000
C	FIND THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	COND1010
	IGH=IG+1	COND1020
	DO 152 11=IGH,IGMAX.	COND1030
	IF(GAPR(11,J).LT.(-1.5E-10)) GO TO 151	COND1040
	GO TO 152	COND1050
C	EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT THE	COND1060
C	HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	COND1070
151	IF(GAPR(11,J).LT.(-2.5E-10)) GO TO 156	COND1080
	IGH=11	COND1090
	GO TO 153	COND1100
152	CONTINUE	COND1110
153	IF(ABS(REMH(IG,J)).GT. .999E6.OR.ABS(REML(IGH,J)).GT. .999E6) GO	COND1120
	TO 154	COND1130
	IF(ABS(T(I,J)-T(I+1,J)).LT.TDMIN) GO TO 155	COND1140
	RAD=.1713E-8*(RDBTH(IG,J)**4-RDBTL(IGH,J)**4)/	COND1150
	*(REMH(IG,J)+REML(IGH,J)-1.0)	COND1160
	HX=CONR(I+1,J)*(T(I,J)-T(I+1,J)-RAD*CONR(I,J)*DEL(RI)*RLN(I))/	COND1170
	1*(T(I,J)-T(I+1,J)+RAD*CONR(I+1,J))	COND1180
	MATXS(IG,J)=200	COND1190
	MATXS(IGH,J)=200	COND1200
	GO TO 157	COND1210
154	MATXS(IG,J)=100	COND1220
	MATXS(IGH,J)=100	COND1230
	GO TO 155	COND1240
155	MATXS(IG,J)=300	COND1250
	MATXS(IGH,J)=300	COND1260
156	HX=CONR(I+1,J)	COND1270
157	KR(I,J)=1.0/(CONR(I,J)*RLN(I)+HX*RATOC(IG))	COND1280
	GO TO 180	COND1290
C	ARE TWO COOLANTS ADJACENT? -'NO,YES'-	COND1300
160	IF (GAPR(IG,J).GT.(-2.5E-10)) GO TO 170	COND1310
C	DO NOT ALLOW HEAT TRANSFER BETWEEN COOLANTS	COND1320
	KR(I,J)=1.0E-10	COND1330
	GO TO 180	COND1340
C	CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE	COND1350
C	LOW INDEX SIDE.	COND1360
C	EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.	COND1370

```

170 IF(IG.EQ.1) GO TO 177                                COND1380
C FIND THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK. COND1390
  IG1=IG-1                                                COND1400
  DO 172 I1=1,IG1                                         COND1410
  IG2=IG-I1                                                COND1420
  IF(GAPR(IG2,J).LT.(-.5E-10)) GO TO 171                 COND1430
  GO TO 172                                                COND1440
C EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT THE COND1450
C LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.     COND1460
171 IF(GAPR(IG2,J).LT.(-2.5E-10)) GO TO 177              COND1470
  IGL=IG2                                                  COND1480
  GO TO 173                                                COND1490
172 CONTINUE                                              COND1500
173 IF(MATRG(IG,J).EQ.100.OR.MATRG(IG,J).EQ.300) GO TO 177 COND1510
  IF(ABS((T(I,J)-T(I+1,J)).LT.TDMIN) GO TO 174           COND1520
  RAD=0.1713E-8*(RBBTH(IGL,J)**4-RBRTL(IG,J)**4)/      COND1530
  *(REMH(IGL,J)+REML(IG,J)-1.0)                         COND1540
  HX=CONR(I,J)*(T(I+1,J)-T(I,J))+CONR(I+1,J)*RAD*RATIOB(IG,J)*DELR(I) COND1550
  1/RATIOK(IG))/(T(I+1,J)-T(I,J)-CONR(I,J)*RAD*RATIOB(IG,J)) COND1560
  GO TO 178                                                COND1570
174 MATRG(IG,J)=300                                       COND1580
  MATRG(IGL,J)=300                                        COND1590
C RECOVER DATA STORED IN THE REM-RBBT ARRAYS.           COND1600
  CONL=REML(IGL,J)                                        COND1610
  CONH=RBBTL(IGL,J)                                       COND1620
  DO 175 IX=1,IM                                          COND1630
  IF(IGR(IX).NE.IGL) GO TO 175                            COND1640
  IPL=IX                                                  COND1650
  GO TO 176                                                COND1660
175 CONTINUE                                              COND1670
176 HX=CONH                                                COND1680
  KR(IPL,J)=1.0/(CONL*RLN(IPL)+HX*RATIOC(IGL))          COND1690
177 HX=CONR(I,J)                                           COND1700
178 KR(I,J)=1.0/(CONR(I+1,J)/RATIOK(IG)+HX*RATIOC(IG))  COND1710
  IF(MATRG(IG,J).NE.300) GO TO 180                       COND1720
  IF(NOMES) GO TO 180                                     COND1730
  IERROR(1)=IERROR(1)+1                                  COND1740
  WRITE(6,1000) ICOND(1),NNITER,ICOND(2),IGL,ICON(3),IG,ICOND(4),J COND1750
CE *****COND1760
180 CONTINUE                                              COND1770
190 CONTINUE                                              COND1780
200 CONTINUE                                              COND1790
CE *****COND1800
C= =====COND1810
C= =====COND1820
CB CALCULATE THE AVERAGE AXIAL-Y-THETA THERMAL CONDUCTIVITIES BETWEENCOND1830

```

```

C .POINTS CON01840
C CALCULATE THE AXIAL-Y-THETA CONDUCTIVITIES ACROSS EACH CON01850
C .AXIAL-Y-THETA GRIDLINE CON01860
C= =====CON01870
C DO 310 J=1,JM CON01880
C JG=JGZ(J) CON01890
C= =====CON01900
C CONSIDER THE POINTS ALONG THE GRIDLINE CON01910
C DO 300 I=2,IM CON01920
C DOES THE GRIDLINE HAVE ANYWHERE A GAP OR A COOLANT ADJACENT TO IT CON01930
C .? -'NO,YES'- CON01940
C IF (JG,ZB,0) GO TO 240 CON01950
C= =====CON01960
C IS A GAP A COOLANT ADJACENT AT THIS POINT? -'NO,YES'- CON01970
C IF (GAPZ(JG,I).EQ.0.0) GO TO 240 CON01980
C IS A GAP OR A COOLANT ADJACENT TO THE AXIAL-Y-THETA GRIDLINE?-'NO,YES'- CON01990
C .L.,GAP'- CON02000
C IF (GAPZ(JG,I).LT.0.0) GO TO 260 CON02010
C= =====CON02020
CB CALCULATE THE AVERAGE CONDUCTIVITY WITH A GAP PRESENT CON02030
C V1=VATZ(JG,I) CON02040
C TH=ZBBTH(JG,I) CON02050
C TB=ZBBTL(JG,I) CON02060
C FTR=RP(I) CON02070
C FTZ=TP(I) CON02080
C SET GAP WIDTH FOR RECTANGULAR OR CYLINDRICAL GEOMETRY. CON02090
C GWID=GAPZ(JG,I) CON02100
C DXHS=ZAT10H(JG) CON02110
C CONVERT GWID AND DXHS FROM RADIANs TO FEET FOR CIRCULAR GEOMETRY. CON02120
C IF (ISHAPE.EQ.2) GO TO 230 CON02130
C GWID=GWID*RP(I) CON02140
C DXHS=DXHS*RP(I) CON02150
C CALCULATE THE LOCAL GAS CONDUCTIVITY. CON02160
230 CALL WADATA CON02170
C BX=ZEMH(JG,I)+ZEML(JG,I)-1.0 CON02180
C CX=.1/135-8*(TH**2+TB**2)*(TH+TB)/BX CON02190
C X1=CONZ(1,J)*ZAT10B(JG,I) CON02200
C X2=1./((GK/GWID+CX)*DXHS) CON02210
C X3=CONZ(1,J+1) CON02220
C SAVE THE DATA REQUIRED FOR CALCULATING AXIAL-Y-THETA GAP BOUNDARY CON02230
C .TEMPERATURES BY STORING IT IN THE ZEM=ZBBT LOCATIONS FOR THE CON02240
C .CURRENT GAPLINE. CON02250
C ZEML(JG,I)=CONZ(1,J) CON02260
C ZEMH(JG,I)=CONZ(1,J+1) CON02270
C ZBBTL(JG,I)=BX CON02280
C ZBBTH(JG,I)=GK CON02290

```

```

      KZ(I,J)=1./((X1+X2+X3)/ZATIOK(JG))                                COND2300
      GO TO 290                                                            COND2310
CE *****                                                                COND2320
CE *****                                                                COND2330
C ARE THE CONDUCTIVITIES ON BOTH SIDES OF THE GRID PLANE IDENTICAL COND2340
C ? 'NO,YES'=-                                                            COND2350
240 IF (CONZ(I,J).NE.CONZ(I,J+1)) GO TO 250                             COND2360
C ASSIGN ONE OF THEM AS THE AVERAGE CONDUCTIVITY                       COND2370
      KZ(I,J)=1.0/CONZ(I,J)                                              COND2380
      GO TO 290                                                            COND2390
C CALCULATE THE AVERAGE CONDUCTIVITY WHEN NO GAP OR COOLANT IS        COND2400
C PRESENT                                                                COND2410
250 KZ(I,J)=1.0/(CONZ(I,J+1)+(CONZ(I,J)-CONZ(I,J+1))*ZLN(J))          COND2420
      GO TO 290                                                            COND2430
CB CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ADJACENT          COND2440
C SET THE GEOMETRICAL CONSTANT DZ FOR RECTANGULAR OR CYLINDRICAL      COND2450
C GEOMETRY.                                                              COND2460
260 DZ=DELZ(J)                                                           COND2470
C CONVERT DZ FROM RADIANS TO FEET FOR CIRCULAR GEOMETRY.              COND2480
      IF(ISHAPE.EQ.2) DZ=DZ*RP(I)                                         COND2490
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GRIDPLANE?'NO,YES'=-   COND2500
      IF(GAPZ(JG,I).LT.(-1.5E-10)) GO TO 270                             COND2510
C CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE HIGH       COND2520
C INDEX SIDE.                                                            COND2530
C SET THE GEOMETRICAL CONSTANT DXT FOR RECTANGULAR OR CYLINDRICAL     COND2540
C GEOMETRY.                                                              COND2550
      DXT=ZL(J)-ZP(J)                                                     COND2560
C CONVERT DXT FROM RADIANS TO FEET FOR CIRCULAR GEOMETRY.             COND2570
      IF(ISHAPE.EQ.2) DXT=DXT*RP(I)                                       COND2580
C EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.                   COND2590
      IF(JG.EQ.JGMAX) GO TO 266                                           COND2600
C FIND THE HIGH AXIAL-Y-THETA BOUNDARY OF THE INTERNAL COOLANT        COND2610
C BLOCK.                                                                  COND2620
      JG1=JG+1                                                            COND2630
      DO 262 JJ=JG1,JGMAX                                                 COND2640
      IF(GAPZ(JJ,I).LT.(-1.5E-10)) GO TO 261                             COND2650
      GO TO 262                                                            COND2660
C EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT        COND2670
C THE HIGH AXIAL-Y-THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.      COND2680
261 IF(GAPZ(JJ,I).LT.(-2.5E-10)) GO TO 266                             COND2690
      JGH=JJ                                                                COND2700
      GO TO 263                                                            COND2710
262 CONTINUE                                                            COND2720
263 IF(ABS(ZEMH(JG,I)).GT. .999E6.OR.ABS(ZEML(JGH,I)).GT. .999E6) GO   COND2730
      TO 264                                                                COND2740
      IF(ABS(I(I,J)-I(I,J+1)).LT.TDMIN) GO TO 265                      COND2750

```


	RAD=0.1713E-8*(ZBBTH(JG,I)**4-ZBBTL(JG,I)**4)/	COND2760
	.(ZEMH(JG,I)+ZEML(JG,I)-1.0)	COND2770
	HA=CONZ(I,J+1)*(T(I,J)-T(I,J+1)-RAD*CONZ(I,J)*DXT)/	COND2780
	1(T(I,J)-T(I,J+1)+RAJ*CONZ(I,J+1))	COND2790
	MATZG(JG,I)=200	COND2800
	MATZG(JG,I)=200	COND2810
	GO TO 267	COND2820
264	MATZG(JG,I)=100	COND2830
	MATZG(JG,I)=100	COND2840
	GO TO 266	COND2850
265	MATZG(JG,I)=300	COND2860
	MATZG(JG,I)=300	COND2870
266	HA=CONZ(I,J+1)	COND2880
267	KZ(I,J)=DZ/(CONZ(I,J)*DXT+HA)	COND2890
	GO TO 290	COND2900
C	ARE TWO COOLANTS ADJACENT? -'NO,YES'	COND2910
270	IF (GAPZ(JG,I).GT.(-2.5E-10)) GO TO 280	COND2920
C	DO NOT ALLOW HEAT TRANSFER BETWEEN COOLANTS	COND2930
	KZ(I,J)=1.0E-10	COND2940
	GO TO 290	COND2950
C	CALCULATE THE AVERAGE CONDUCTIVITY WITH A COOLANT ON THE LOW	COND2960
C	INDEX SIDE.	COND2970
C	SET GEOMETRICAL CONSTANT DXT FOR RECTANGULAR OR CIRCULAR	COND2980
C	GEOMETRY.	COND2990
280	DXT=2P(J+1)-ZL(J)	COND3000
C	CONVERT DXT FROM RADIANS TO FEET FOR CIRCULAR GEOMETRY.	COND3010
	IF (SHAPE.EQ.2) DXT=DXT*RP(I)	COND3020
C	EXCLUDE CALCULATION OF RAD FOR EXTERNAL COOLANTS.	COND3030
	IF (JG.EQ.1) GO TO 287	COND3040
C	FIND THE LOW AXIAL-Y-THETA BOUNDARY OF THE INTERNAL COOLANT	COND3050
C	BLOCK.	COND3060
	JG1=JG-1	COND3070
	DO 282 JJ=1,JG1	COND3080
	JG2=JG-JJ	COND3090
	IF (GAPZ(JG2,I).LT.(-.5E-10)) GO TO 281	COND3100
	GO TO 282	COND3110
C	EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT	COND3120
C	THE LOW AXIAL-Y-THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.	COND3130
281	IF (GAPZ(JG2,I).LT.(-2.5E-10)) GO TO 287	COND3140
	JGL=JG2	COND3150
	GO TO 283	COND3160
282	CONTINUE	COND3170
283	IF (MATZG(JG,I).EQ.100.OR.MATZG(JG,I).EQ.300) GO TO 287	COND3180
	IF (ABS(T(I,J)-T(I,J+1)).LT.TDMIN) GO TO 284	COND3190
	RAJ=0.1713E-8*(ZBBTH(JGL,I)**4-ZBBTL(JG,I)**4)/	COND3200
	.(ZEMH(JGL,I)+ZEML(JG,I)-1.0)	COND3210

	HX=CONZ(I,J)*(T(I,J+1)-T(I,J)+CONZ(I,J+1)*RAD*DXT)/	COND3220
	.(T(I,J+1)-T(I,J)-CONZ(I,J)*RAD)	COND3230
	GO TO 283	COND3240
284	MATZG(JG,I)=300	COND3250
	MATZG(JGL,I)=300	COND3260
C	RECOVER DATA STORED IN THE MEM-ZBBT ARRAYS.	COND3270
	CONL=ZEML(JGL,I)	COND3280
	CONH=ZBSIL(JGL,I)	COND3290
	DO 285 JX=1,JM	COND3300
	IF(JGZ(JX).NE.JGL) GO TO 285	COND3310
	JPL=JX	COND3320
	GO TO 286	COND3330
285	CONTINUE	COND3340
286	HX=CONH	COND3350
	GFAC=1.0	COND3360
	IF(1.5*HPE.EQ.2) GFAC=RP(I)	COND3370
	JZP=JELL(JPL)*GFAC	COND3380
	DXTP=(ZL(JPL)-ZP(JPL))*GFAC	COND3390
	KZ(I,JPL)=JZP/(CONL*DXTP+HX)	COND3400
287	HX=CONZ(I,J)	COND3410
288	KZ(I,J)=JZ/(CONZ(I,J+1)*DXT+HX)	COND3420
	IF(MATZG(JG,I).NE.300) GO TO 290	COND3430
	IF(JMES) GO TO 290	COND3440
	IERROR(1)=IERROR(1)+1	COND3450
	WRITE(6,1000) ICOND(5),JNITER,ICOND(6),JGL,ICOND(7),JG,ICOND(8),I	COND3460
CE	*****	COND3470
290	CONTINUE	COND3480
300	CONTINUE	COND3490
310	CONTINUE	COND3500
CE	*****	COND3510
	RETURN	COND3520
	END	COND3530

```

SUBROUTINE STEP(I,NOTEST)                                STFP 10
INCLUDE COMJIM                                           STFP 20
C= =====STFP 30
CB CALCULATE THE NEW TEMPERATURES AFTER ONE TIMESTEP    STFP 40
C= =====STFP 50
C= =====STFP 60
LOGICAL NOTEST                                           STFP 70
DIMENSION X (M0),Y (M0)                                STFP 80
C= =====STFP 90
DO 140 I=2,IM                                           STFP 100
DO 120 J=2,JM                                           STFP 110
XY=0.0                                                  STFP 120
C1=KR(I-1,J)*KR(I-1,J)                                STFP 130
C2=KR(I,J)*KR(I,J)                                    STFP 140
C3=KZ(I,J-1)*KZ(I,J-1)                                STFP 150
C4=KZ(I,J)*KZ(I,J)                                    STFP 160
C5=KCP(I,J)/DT                                          STFP 170
E1=-C5-C3-C4                                           STFP 180
IF (J.NE.2) GO TO 100                                  STFP 190
XY=C4*T(I,1)                                           STFP 200
C3=0.0                                                  STFP 210
GO TO 110                                               STFP 220
100 IF (J.NE.JM) GO TO 110                             STFP 230
XY=C4*T(I,J+1)                                         STFP 240
C4=0.0                                                  STFP 250
110 E=E1-C3*X(J-1)                                     STFP 260
X(J)=C4/E                                              STFP 270
Y(J)=-(C5*T(I,J)+C1*(T(I-1,J)-T(I,J))+C2*(T(I+1,J)-T(I,J))+XY+W(I, STFP 280
J)+C3*Y(J-1))/E                                         STFP 290
120 CONTINUE                                           STFP 300
J=JM                                                  STFP 310
130 TF(I,J)=Y(J)-X(J)*TF(I,J+1)                       STFP 320
J=J-1                                                  STFP 330
IF (J.GT.1) GO TO 130                                  STFP 340
140 CONTINUE                                           STFP 350
C= =====STFP 360
DO 140 J=2,JM                                           STFP 370
DO 170 I=2,IM                                           STFP 380
XY=0.0                                                  STFP 390
C1=KR(I-1,J)*KR(I-1,J)                                STFP 400
C2=KR(I,J)*KR(I,J)                                    STFP 410
C3=KZ(I,J-1)*KZ(I,J-1)                                STFP 420
C4=KZ(I,J)*KZ(I,J)                                    STFP 430
C5=KCP(I,J)/DT                                          STFP 440
E1=-C1-C2-C5                                           STFP 450

```

IF (I.NE.2) GO TO 150	STFP 460
XY=C1*T(I,J)	STFP 470
C1=0.0	STFP 480
GO TO 160	STFP 490
150 IF (I.NE.IM) GO TO 160	STFP 500
XY=C2*T(I+1,J)	STFP 510
C2=0.0	STFP 520
160 E=E1-C1*lambda(I-1)	STFP 530
X(I)=C2/E	STFP 540
Y(I)=-((C3*T(I,I,J)+C3*(TT(I,J-1)-TT(I,J))+C4*(TT(I,J+1)-TT(I,J))+XY	STFP 550
1+W(I,J)+C1*Y(I-1))/E	STFP 560
170 CONTINUE	STFP 570
I=IM	STFP 580
180 T(I,J)=Y(I)-X(I)*T(I+1,J)	STFP 590
I=I-1	STFP 600
IF (I.GT.1) GO TO 180	STFP 610
190 CONTINUE	STFP 620
IF (.NOT.SW(10).OR.NOTEST) GO TO 210	STFP 630
DO 200 I=2,IM	STFP 640
DO 200 J=2,JM	STFP 650
IF (T(I,J).LT.0.0.OR.T(I,J).GT.1.0E6) RETURN 1	STFP 660
200 CONTINUE	STFP 670
210 RETURN	STFP 680
C=	STFP 690
END	STFP 700

```

SUBROUTINE COOL                                COOL 10
INCLUDE COMDIM                                COOL 20
=====COOL 30
C CALCULATE THE COOLANT TEMPERATURES          COOL 40
C=====COOL 50
C ERROR MESSAGE                               COOL 60
C      COOL1, COOL2, COOL3                    COOL 70
C      THERE IS A PROGRAM LIMITATION WHICH MAY BE ENCOUNTERED COOL 80
C      WHEN CALCULATING COOLANT TEMPERATURES IN THE PRESENCE COOL 90
C      OF RADIATION BETWEEN COOLANT BLOCK BOUNDARIES. IT COOL 100
C      ARISES WHEN A COOLANT POINT AND AN ADJACENT MATERIAL COOL 110
C      POINT APPROACH THE SAME TEMPERATURE. THESE MESSAGES COOL 120
C      INDICATE THAT THIS LIMITATION HAS BEEN ENCOUNTERED COOL 130
C      AND CIRCUMVENTED BY ASSIGNING ZERO COOLANT HEAT COOL 140
C      TRANSFER AT THE LEVEL INDICATED. BY USING A SMALLER COOL 150
C      TIME STEP, THIS DIFFICULTY CAN USUALLY BE ELIMINATED. COOL 160
C      COOL1 HEAT TRANSFER WITHIN AN AXIAL-Y-THETA COOLANT COOL 170
C      IS ZERO IN BLOCK I AT AXIAL-Y-THETA POINT COOL 180
C      LEVEL J. COOL 190
C      COOL2 HEAT TRANSFER WITHIN A RADIAL-X COOLANT IS COOL 200
C      ZERO IN BLOCK L AT RADIAL-X POINT LEVEL J. COOL 210
C      COOL3 HEAT TRANSFER WITHIN A NORMAL COOLANT IS ZERO COOL 220
C      IN BLOCK L. COOL 230
C=====COOL 240
C=====COOL 250
C=====COOL 260
C LOGICAL ANTI, NOMES, IEP, IRAD, IXRAD COOL 270
C DIMENSION IC (MQ), ICNOL (5) COOL 280
C=====COOL 290
C=====COOL 300
C=====COOL 310
C=====COOL 320
C=====COOL 330
C=====COOL 340
C=====COOL 350
C=====COOL 360
C=====COOL 370
C=====COOL 380
C=====COOL 390
C=====COOL 400
C=====COOL 410
C IS AT A COOLANT BLOCK -NO, YES'- COOL 420
C IF (MB(L).GE.0) GO TO 500 COOL 430
C NE=MB(L) COOL 440
C TIN=TI(N) COOL 450

```

C	DEFINE THE CORNERS OF THE COOLANT BLOCK	COOL 460
	ILSEL(L)	COOL 470
	JLSEL(L)	COOL 480
	IMSEL(L)	COOL 490
	JMSEL(L)	COOL 500
C	FIND THE HIGH BOUNDING GAPLINES OF THE COOLANT BLOCK.	COOL 510
	IF(IHS.EQ.IMAX) GO TO 90	COOL 520
	IHS=IGR(IHS)	COOL 530
	GO TO 93	COOL 540
90	IGHSENRO	COOL 550
93	IF(JHS.EQ.JMAX) GO TO 95	COOL 560
	JHS=JGZ(JHS)	COOL 570
	GO TO 97	COOL 580
95	JHS=JZ0	COOL 590
C	DETERMINE THE DIRECTION OF COOLANT FLOW -'RADIAL-X,AXIAL-Y-THETA,NORMAL'	COOL 600
C	.NORMAL'	COOL 610
97	ANTI=.FALSE.	COOL 620
	IF (IPATH(N).LT.0) ANTI=.TRUE.	COOL 630
	I=IASS(IPATH(N))	COOL 640
	GO TO (240,100,370), I	COOL 650
C=	=====	COOL 660
C	CALCULATE THE AXIAL-Y-THETA COOLANT TEMPERATURES	COOL 670
C	ASSIGN THE INLET TEMPERATURE	COOL 680
100	TC(JLS)=TIN	COOL 690
	IF (ANTI) TC(JHS)=TIN	COOL 700
C	CALCULATE THE COOLANT TEMPERATURES ALONG THE FLOW DIRECTION	COOL 710
	DO 230 J=JLS,JHS	COOL 720
	J=J1	COOL 730
	IF (ANTI) J=JHS+JLS-J	COOL 740
	IFP=.FALSE.	COOL 750
	IRAP=.FALSE.	COOL 760
	IF(MATRO(IGH5,J).EQ.200) IRAP=.TRUE.	COOL 770
	SUMKA=0.0	COOL 780
	SUMKAT=0.0	COOL 790
C	CALCULATE THE HEATTRANSFER ACROSS THE COOLANT BOUNDARIES AT THIS	COOL 800
C	.LEVEL	COOL 810
C	SUM THE HEAT TRANSFER ACROSS THE RADIAL-X BOUNDARIES OF THE	COOL 820
C	.COOLANT	COOL 830
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX SIDE	COOL 840
C	X='YES,NO'	COOL 850
	IF (ILS.LE.2) GO TO 120	COOL 860
C	SUM THE HEAT TRANSFER ACROSS THE LOW RADIAL-X BOUNDARY	COOL 870
	I=ILS-1	COOL 880
	X=RX(I,J)*XR(I,J)	COOL 890
	SUMKAT=SUMKAT+X*T(I,J)	COOL 900
	SUMKA=SUMKA+X	COOL 910

	IF (IRAD)	COOL 920
	.CALL RANGE(IFP,T(I,J),THI,TLO)	COOL 930
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX SIDE	COOL 940
C	X='YES,NO'	COOL 950
120	IF (IHS.GE.IM) GO TO 130	COOL 960
C	SUM THE HEAT TRANSFER ACROSS THE HIGH RADIAL-X BOUNDARY	COOL 970
	I=IHS	COOL 980
	X=KK(1,J)*KK(I,J)	COOL 990
	SUMKA1=SUMKAT+X*T(I+1,J)	COOL 1000
	SUMKA=SUMKA+X	COOL 1010
	IF (IRAD)	COOL 1020
	.CALL RANGE(IFP,T(I+1,J),THI,TLO)	COOL 1030
C	HAS THE HEAT TRANSFER THROUGH ALL RADIAL-X POINTS BEEN CALCULATED?	COOL 1040
C	X='YES,NO'	COOL 1050
130	CONTINUE	COOL 1060
CE	*****	COOL 1070
C/	CALCULATE THE COOLANT TEMPERATURES AT THIS LEVEL	COOL 1080
	WCP=FLOW(N)*RCPC(N,J)	COOL 1090
	EX=SUMKA/WCP	COOL 1100
C	IS THE RADIATION EFFECT UPON THE HEAT TRANSFER COEFFICIENT TOO	COOL 1110
C	LARGE? -'NO,YES'	COOL 1120
	IF (EX.GT.-30.) GO TO 150	COOL 1130
	GO TO 185	COOL 1140
150	EXPO=EXP(-EX)	COOL 1150
C	CALCULATE THE COOLANT TEMPERATURE AT THE END OF THE INCREMENT	COOL 1160
	X=TC(J)+EXPO*SUMKAT/SUMKA*(1.0-EXPO)	COOL 1170
C	CALCULATE THE AVERAGE COOLANT TEMPERATURE OF THE CURRENT LEVEL	COOL 1180
	TAV=(SUMKA1-WCP*(X-TC(J)))/SUMKA	COOL 1190
C	DOES THE MEAN TEMPERATURE FALL BETWEEN THE INLET AND EXIT	COOL 1200
C	TEMPERATURES ?	COOL 1210
	IF (TC(J).GT.X) GO TO 170	COOL 1220
	IF (TAV.GE.TC(J).AND.TAV.LE.X) GO TO 175	COOL 1230
	GO TO 173	COOL 1240
170	IF (TAV.GE.X.AND.TAV.LE.TC(J)) GO TO 175	COOL 1250
173	TAV=(TC(J)+X)*0.5	COOL 1260
C	DOES THE EXIT TEMPERATURE FALL WITHIN THE LIMITS IMPOSED BY THE	COOL 1270
C	INLET TEMPERATURE AND THE ADJACENT MATERIAL TEMPERATURES ?	COOL 1280
175	IF (.NOT.IRAJ) GO TO 190	COOL 1290
	IF (TC(J).GT.THI) THI=TC(J)	COOL 1300
	IF (TC(J).LT.TLO) TLO=TC(J)	COOL 1310
	IF (X.LE.THI.AND.X.GE.TLO) GO TO 190	COOL 1320
185	X=TC(J)	COOL 1330
	TAV=TC(J)	COOL 1340
	IF (NMES) GO TO 190	COOL 1350
	IERROR(1)=IERROR(1)+1	COOL 1360
	WRITE(6,1000) ICOOL(1),NNITER,L,ICOOL(4),J	COOL 1370

190	IF (ANT1) GO TO 200	COOL1380
	TC(J+1)=X	COOL1390
	GO TO 210	COOL1400
200	TC(J-1)=X	COOL1410
CE	*****	COOL1420
C	RECORD THE COOLANT TEMPERATURE OF THE CURRENT LEVEL	COOL1430
210	DO 220 I=ILS,IHS	COOL1440
	T(I,J)=TAV	COOL1450
	TT(I,J)=TAV	COOL1460
220	CONTINUE	COOL1470
C	HAVE ALL LEVELS BEEN CALCULATED-'YES,NO'-	COOL1480
230	CONTINUE	COOL1490
C	ASSIGN THE COOLANT OUTLET TEMPERATURE	COOL1500
	TO(N)=X	COOL1510
	GO TO 500	COOL1520
CE	*****	COOL1530
C	CALCULATE THE RADIAL-X COOLANT TEMPERATURES	COOL1540
C	ASSIGN THE INLET TEMPERATURE	COOL1550
240	TC(ILS)=TIN	COOL1560
	IF (ANT1) TC(IHS)=TIN	COOL1570
C	CALCULATE THE COOLANT TEMPERATURES ALONG THE FLOW DIRECTION	COOL1580
	DO 300 I=ILS,IHS	COOL1590
	I=I1	COOL1600
	IF (ANT1) I=IHS+ILS-I	COOL1610
	IFPE=FALSE.	COOL1620
	IRAD=FALSE.	COOL1630
	IF (M1Z0(JHS,I).EQ.200) IRAD=TRUE.	COOL1640
	SUMKA=0.0	COOL1650
	SUMKAT=0.0	COOL1660
CB	CALCULATE THE HEAT TRANSFER ACROSS THE COOLANT BOUNDARIES AT THIS	COOL1670
C	LEVEL	COOL1680
C	SUM THE HEAT TRANSFER ACROSS THE AXIAL-Y-THETA BOUNDARIES OF THE	COOL1690
C	COOLANT	COOL1700
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX SIDE	COOL1710
C	A-'YES,NO'-	COOL1720
	IF (JLS.LE.2) GO TO 260	COOL1730
C	SUM THE HEAT TRANSFER ACROSS THE LOW AXIAL-Y-THETA BOUNDARY	COOL1740
	J=JLS-1	COOL1750
	X=RZ(I,J)*KZ(I,J)	COOL1760
	SUMKAT=SUMKAT+X*T(I,J)	COOL1770
	SUMKA=SUMKA+X	COOL1780
	IF (IRAD)	COOL1790
	.CALL RANGE(IFP,T(I,J),THI,TLO)	COOL1800
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX SIDE	COOL1810
C	A-'YES,NO'-	COOL1820
260	IF (JHS.GE.JM) GO TO 270	COOL1830


```

C      SUM THE HEAT TRANSFER ACROSS THE HIGH AXIAL-Y-THETA BOUNDARY      C00L1840
      J=JHS                                                                C00L1850
      X=RZ(1,J)*KZ(I,J)                                                  C00L1860
      SUMKAT=SUMKAT+X*T(I,J+1)                                          C00L1870
      SUMKA=SUMKA+X                                                       C00L1880
      IF(1-RAJ)                                                           C00L1890
        .CALL RANGE(1,FP,T(I,J+1),TH1,TLO)                             C00L1900
      HAS THE HEAT TRANSFER THROUGH ALL AXIAL-Y-THETA POINTS BEEN CALCUL C00L1910
C      .ATEUP='YES,NO'=-
C      270 CONTINUE                                                       C00L1920
C/  CALCULATE THE COOLANT TEMPERATURES AT THIS LEVEL                    C00L1930
      WCP=FLOW(N)*RCPC(N,1)                                              C00L1940
      EX=SUMKA/WCP                                                         C00L1950
C      IS THE RADIATION EFFECT UPON THE HEAT TRANSFER COEFFICIENT TOO   C00L1960
C      .LARGE? -'NO,YES'=-
      IF (EX.GT.-.50.) GO TO 280                                         C00L1970
      GO TO 315                                                            C00L1980
      280 EXP0=EXP(-EX)                                                    C00L1990
C      CALCULATE THE COOLANT TEMPERATURE AT THE END OF THE INCREMENT     C00L2000
      X=TC(1)*EXP0+SUMKAT/SUMKA*(1.0-EXP0)                             C00L2010
C      CALCULATE THE AVERAGE COOLANT TEMPERATURE OF THE CURRENT LEVEL    C00L2020
      TAV=(SUMKAT-WCP*(X-TC(1)))/SUMKA                                  C00L2030
C      DOES THE MEAN TEMPERATURE FALL BETWEEN THE INLET AND EXIT         C00L2040
C      TEMPERATURES ?
      IF (TC(1).GT.X) GO TO 300                                           C00L2050
      IF (TAV.GE.IC(I).AND.TAV.LE.X) GO TO 310                           C00L2060
      GO TO 305                                                            C00L2070
      300 IF (TAV.GE.X.AND.TAV.LE.TC(1)) GO TO 310                       C00L2080
      305 TAV=(TC(1)+X)*0.5                                               C00L2090
C      DOES THE EXIT TEMPERATURE FALL WITHIN THE LIMITS IMPOSED BY THE   C00L2100
C      INLET TEMPERATURE AND THE ADJACENT MATERIAL TEMPERATURES ?
      310 IF(.NOT.1RAJ) GO TO 320                                         C00L2110
      IF(IC(I).GT.THI) TH1=TC(I)                                          C00L2120
      IF(IC(I).LT.TLO) TLO=TC(I)                                          C00L2130
      IF(X.LE.THI.AND.X.GE.TLO) GO TO 320                                C00L2140
      315 X=TC(I)                                                         C00L2150
      TAV=TC(1)                                                           C00L2160
      IF(NOMES) GO TO 320                                                 C00L2170
      IERROR(1)=IERROR(1)+1                                              C00L2180
      WRITE(6,1000) IC00L(2),NNITER,L,IC00L(5),I                      C00L2190
      320 IF (ANT1) GO TO 330                                             C00L2200
      TC(1+1)=X                                                           C00L2210
      GO TO 340                                                            C00L2220
      330 TC(1-1)=X                                                       C00L2230
CE *****C00L2240
C      RECORD THE COOLANT TEMPERATURE OF THE CURRENT LEVEL              C00L2250
      C00L2260
      C00L2270
      C00L2280
      C00L2290

```

340	DO 350 J=JLS,JHS	COOL2300
	T(I,J)=TAV	COOL2310
	TT(I,J)=TAV	COOL2320
350	COOL2330	COOL2330
C	HAVE ALL LEVELS BEEN CALCULATED='YES,NO'=-	COOL2340
360	COOL2350	COOL2350
C	ASSIGN THE COOLANT OUTLET TEMPERATURE	COOL2360
	TO(I)=X	COOL2370
	GO TO 500	COOL2380
CE	*****	COOL2390
CB	CALCULATE THE NORMAL COOLANT TEMPERATURES	COOL2400
C	ASSIGN THE INLET TEMPERATURE	COOL2410
370	TC(1)=TIN	COOL2420
	SUMKA=0.0	COOL2430
	SUMKAT=0.0	COOL2440
	IFP=.FALSE.	COOL2450
	IXRAD=.FALSE.	COOL2460
CB	CALCULATE THE HEATTRANSFER ACROSS THE COOLANT BOUNDARIES	COOL2470
C	SUM THE HEATTRANSFER ACROSS THE RADIAL-X BOUNDARIES OF THE COOLANT	COOL2480
	DO 410 J=JLS,JHS	COOL2490
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX	COOL2500
C	SIDE='YES,NO'=-	COOL2510
	IF (IHS.LE.2) GO TO 400	COOL2520
C	SUM THE HEATTRANSFER ACROSS THE LOW RADIAL-X BOUNDARY	COOL2530
	I=IHS-1	COOL2540
	IRAD=.FALSE.	COOL2550
	IF (MATRG(IHS,J).EQ.200) IRAD=.TRUE.	COOL2560
	IF (IRAD) IXRAD=.TRUE.	COOL2570
	X=RX(I,J)*KR(I,J)	COOL2580
	SUMKAT=SUMKAT+X*T(I,J)	COOL2590
	SUMKA=SUMKA+X	COOL2600
	IF (IRAD)	COOL2610
	CALL RANGE(IFP,T(I,J),THI,TLO)	COOL2620
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX	COOL2630
C	SIDE='YES,NO'=-	COOL2640
400	IF (IHS.GE.1M) GO TO 410	COOL2650
C	SUM THE HEATTRANSFER ACROSS THE HIGH RADIAL-X BOUNDARY	COOL2660
	I=IHS	COOL2670
	IRAD=.FALSE.	COOL2680
	IF (MATRG(IHS,J).EQ.200) IRAD=.TRUE.	COOL2690
	IF (IRAD) IXRAD=.TRUE.	COOL2700
	X=RX(I,J)*KR(I,J)	COOL2710
	SUMKAT=SUMKAT+X*T(I+1,J)	COOL2720
	SUMKA=SUMKA+X	COOL2730
	IF (IRAD)	COOL2740
	CALL RANGE(IFP,T(I+1,J),THI,TLO)	COOL2750

C	HAS THE HEATTRANSFER THROUGH ALL RADIAL-X POINTS BEEN	CONL2760
C	.CALCULATED? -'YES,NO'-	CONL2770
410	CONTINUE	CONL2780
C	SUM THE HEAT TRANSFER ACROSS THE AXIAL-Y-THETA BOUNDARIES OF THE	CONL2790
C	.COOLANT	CONL2800
	DO 450 I=ILS,IHS	CONL2810
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE LOW INDEX	CONL2820
C	.SIDE? -'YES,NO'-	CONL2830
	IF(JLS.LE.2) GO TO 440	CONL2840
C	SUM THE HEATTRANSFER ACROSS THE LOW AXIAL-Y-THETA BOUNDARY	CONL2850
	J=JLS-1	CONL2860
	IRAD=.FALSE.	CONL2870
	IF(MATZ(JHS,I).EQ.200) IRAD=.TRUE.	CONL2880
	IF(IRAD) IXRAD=.TRUE.	CONL2890
	X=RZ(I,J)*KZ(I,J)	CONL2900
	SUMKAT=SUMKAT+X*T(I,J)	CONL2910
	SUMKA=SUMKA+X	CONL2920
	IF(1RAD)	CONL2930
	.CALL RANGE(IFP,T(I,J),THI,TLO)	CONL2940
C	IS THE COOLANT ADJACENT TO THE OUTSIDE ON THE HIGH INDEX	CONL2950
C	.SIDE? -'YES,NO'-	CONL2960
440	IF (JHS.GE.JM) GO TO 450	CONL2970
C	SUM THE HEATTRANSFER ACROSS THE HIGH AXIAL-Y-THETA BOUNDARY	CONL2980
	J=JHS	CONL2990
	IRAD=.FALSE.	CONL3000
	IF(MATZ(JHS,I).EQ.200) IRAD=.TRUE.	CONL3010
	IF(IRAD) IXRAD=.TRUE.	CONL3020
	X=RZ(I,J)*KZ(I,J)	CONL3030
	SUMKAT=SUMKAT+X*T(I,J+1)	CONL3040
	SUMKA=SUMKA+X	CONL3050
	IF(1RAD)	CONL3060
	.CALL RANGE(IFP,T(I,J+1),THI,TLO)	CONL3070
C	HAS THE HEATTRANSFER THROUGH ALL AXIAL-Y-THETA POINTS BEEN CALCULATED?	CONL3080
C	.TEJP? -'YES,NO'-	CONL3090
450	CONTINUE	CONL3100
C/	CALCULATE THE COOLANT TEMPERATURE	CONL3110
	WCP=FLOW(N)*RCPC(N,1)	CONL3120
	EX=SUMKA/WCP	CONL3130
C	IS THE RADIATION EFFECT UPON THE HEATTRANSFER COEFFICIENT TOO	CONL3140
C	.LARGE? -'NO,YES'-	CONL3150
	IF (EX.GT.-50.) GO TO 460	CONL3160
	GO TO 483	CONL3170
460	EXPO=EXP(-EX)	CONL3180
C	CALCULATE THE OUTLET COOLANT TEMPERATURE	CONL3190
	X=TC(1)*EXPO+SUMKAT/SUMKA*(1.0-EXPO)	CONL3200
C	CALCULATE THE AVERAGE COOLANT TEMPERATURE	CONL3210

```

470 TAV=(SUMKAI-KCP*(X-TC(1)))/SUMKA                                C00L3220
C    DOES THE MEAN TEMPERATURE FALL BETWEEN THE INLET AND EXIT      C00L3230
C    TEMPERATURES ?                                                  C00L3240
    IF (IC(1).GT.X) GO TO 475                                         C00L3250
    IF (TAV.GE.TC(1).AND.TAV.LE.X) GO TO 480                         C00L3260
    GO TO 477                                                         C00L3270
475 IF (TAV.GE.X.AND.TAV.LE.TC(1)) GO TO 480                         C00L3280
477 TAV=(TC(1)+X)*0.5                                                C00L3290
C    DOES THE EXIT TEMPERATURE FALL WITHIN THE LIMITS IMPOSED BY THE C00L3300
C    INLET TEMPERATURE AND THE ADJACENT MATERIAL TEMPERATURES ?    C00L3310
480 IF (.NOT.IXRAD) GO TO 485                                         C00L3320
    IF (IC(1).GT.THI) THI=TC(1)                                       C00L3330
    IF (IC(1).LT.TLO) TLO=TC(1)                                       C00L3340
    IF (X.LE.THI.AND.X.GE.TLO) GO TO 485                             C00L3350
483 X=TC(1)                                                          C00L3360
    TAV=TC(1)                                                         C00L3370
    IF (NOMES) GO TO 485                                              C00L3380
    IERROR(1)=IERROR(1)+1                                           C00L3390
    WRITE(6,1000) IC00L(3),NITER,L                                  C00L3400
485 TC(2)=X                                                         C00L3410
    DO 490 I=ILS,IHS                                                C00L3420
    DO 490 J=JLS,JHS                                                C00L3430
    T(I,J)=TAV                                                       C00L3440
490 CONTINUE                                                         C00L3450
    TO(N)=X                                                           C00L3460
CE *****/*****C00L3470
C    HAVE ALL BLOCKS BEEN CHECKED -YES,NO?-                         C00L3480
500 CONTINUE                                                         C00L3490
    RETURN                                                            C00L3500
C=====C00L3510
C *****C00L3520
C *****C00L3530
C *****C00L3540
    SUBROUTINE RANGE(IFP,TX,THI,TLO)                                C00L3550
C=====C00L3560
C    DETERMINE THE CURRENT VALUES OF THE HIGHEST AND LOWEST ADJACENT C00L3570
C    MATERIAL POINT TEMPERATURES.                                  C00L3580
C=====C00L3590
    LOGICAL IFP                                                       C00L3600
C=====C00L3610
    IF (IFP) GO TO 100                                                C00L3620
    THI=TX                                                            C00L3630
    TLO=TX                                                            C00L3640
    IFP=.TRUE.                                                         C00L3650
    GO TO 110                                                         C00L3660
100 IF (IX.GT.THI) THI=TX                                             C00L3670
    IF (IX.LT.TLO) TLO=TX                                             C00L3680
C=====C00L3690
110 RETURN                                                            C00L3700
    END                                                                C00L3710

```

```

SUBROUTINE SURT
  INCLUDE COMDIM
  ===== SURT 10
  C= ===== SURT 20
  C CALCULATE THE BOUNDARY TEMPERATURES ASSOCIATED WITH THE POINT SURT 30
  C . TEMPERATURE RESULTS OF THE CURRENT ITERATION. SURT 40
  C= ===== SURT 50
  C ERROR STOPS= SURT 60
  C SURT 1 INSTABILITY ENCOUNTERED WHILE CALCULATING THE SURT 70
  C RADIAL-X BOUNDARY TEMPERATURES OF A GAP. USE A SURT 80
  C SMALLER TIME STEP. SURT 90
  C SURT 2 INSTABILITY ENCOUNTERED WHILE CALCULATING THE SURT 100
  C AXIAL-Y-THETA BOUNDARY TEMPERATURES OF A GAP. SURT 110
  C USE A SMALLER TIME STEP. SURT 120
  C ===== SURT 130
  C= ===== SURT 140
  C DIMENSION ISURT(3) SURT 150
  C DATA(ISURT(1),I=1,3) /OHSURT , 1H1, 1H2/ SURT 160
  C ===== SURT 170
  C IGMAX=IGR(IM) SURT 180
  C JGMAX=JGZ(JM) SURT 190
  C TDMIN=0.10 SURT 200
  C VNITER=VNITER+1 SURT 210
  C ===== SURT 220
  C CALCULATE THE BOUNDARY TEMPERATURES ADJACENT GAPS AND COOLANTS. SURT 230
  C ===== SURT 240
  C CALCULATE THE RADIAL-X BOUNDARY TEMPERATURES. SURT 250
  C DO 210 I=1,14 SURT 260
  C IG=IGR(I) SURT 270
  C ===== SURT 280
  C EXCLUDE THE NON-GAP LINES. SURT 290
  C IF(IG.EQ.0)GO TO 210 SURT 300
  C ===== SURT 310
  C CONSIDER THE POINTS ALONG EACH RADIAL-X GAP LINE. SURT 320
  C DO 200 J=2,J4 SURT 330
  C IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAP LINE AT THIS SURT 340
  C POINT ? -'NO,YES'- SURT 350
  C IF(GAPR(IG,J).EQ.0.0)GO TO 200 SURT 360
  C IS IT A COOLANT OR A GAP WHICH IS ADJACENT ? -'COOLANT,GAP'- SURT 370
  C IF(GAPR(IG,J).LT.0.0)GO TO 130 SURT 380
  C ===== SURT 390
  C CALCULATE THE BOUNDARY TEMPERATURES WHEN A GAP IS ADJACENT. SURT 400
  C RECOVER DATA STORED IN THE REM-RBBT ARRAYS. SURT 410
  C BX=RBBTL(IG,J) SURT 420
  C GK=RBBTH(IG,J) SURT 430
  C CONL=REML(IG,J) SURT 440
  C ===== SURT 450

```

	CONH=REM4(IG,J)	SURT 460
C	GET READY TO CALCULATE THE BOUNDARY TEMPERATURES OF THE GAP.	SURT 470
	B=RATIOB(IG,J)*CONL/CONH	SURT 480
	C=1(I,J)+3*I(I+1,J)	SURT 490
	RGAP=0.1713E-8*GAPR(IG,J)/BX	SURT 500
	ED=KALIOH(IG)*CONH/GAPR(IG,J)	SURT 510
C	ALLOW 10 ITERATIONS TO CALCULATE THE BOUNDARY TEMPERATURES.	SURT 520
	DO 120 KK=1,10	SURT 530
	IF(KK.EQ.1)GO TO 100	SURT 540
	TH=K3BT(IG,J)	SURT 550
	TB=K3BT(IG,J)	SURT 560
	GO TO 110	SURT 570
100	TH=1(I,J)	SURT 580
	TB=1(I+1,J)	SURT 590
110	GH=(GX+RGAP*(TH**2+TB**2)*(TH+TB))*E7	SURT 600
	X1=GH*C	SURT 610
	X2=1.0+GH*(B+1.0)	SURT 620
	ROBIL(IG,J)=(X1+1(I+1,J))/X2	SURT 630
	ROBIN(IG,J)=(X1+T(I,J))/X2	SURT 640
C	ARE THE BOUNDARY TEMPERATURES CONVERGED ? -'YES,NO'-	SURT 650
	IF(ABS(K3BT(IG,J)-TH).LE.1.0)GO TO 500	SURT 660
C	HAVE THE 10 ITERATIONS BEEN PERFORMED ? -'YES,NO'-	SURT 670
120	CONTINUE	SURT 680
C	NEGATIVE TEMPERATURES ARE PRESENT AND RESULT IN LOCAL	SURT 690
C	INSTABILITIES. TRY A SMALLER TIME STEP.	SURT 700
	ERROR(1)=ISURT(1)	SURT 710
	ERROR(2)=ISURT(2)	SURT 720
	CALL ERROR2	SURT 730
C	=====	SURT 740
C	CALCULATE THE BOUNDARY TEMPERATURES WHEN A COOLANT IS ADJACENT.	SURT 750
C	IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE ? -'NO,YES'-	SURT 760
130	IF(GAPR(IG,J).LT.(-1.5E-10))GO TO 160	SURT 770
C	CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE HIGH	SURT 780
C	INDEX SIDE.	SURT 790
C	RECOVER DATA STORED IN THE REM-RBBT ARRAYS.	SURT 800
	CONLEREM(IG,J)	SURT 810
	CONHERBUTL(IG,J)	SURT 820
C	EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.	SURT 830
	IF(IG.EQ.IGMAX) GO TO 140	SURT 840
	IF(MATRG(IG,J).NE.200) GO TO 140	SURT 850
C	FIND THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT 860
	IG1=IG+1	SURT 870
	DO 134 II=IG1,IGMAX	SURT 880
	IF(GAPR(II,J).LT.(-1.5E-10)) GO TO 132	SURT 890
	GO TO 134	SURT 900
C	EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT	SURT 910

C	THE HIGH RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT 920
132	IF(GAPR(II,J).LT.(-2.5E-10)) GO TO 140	SURT 930
	IGH=II	SURT 940
	GO TO 138	SURT 950
134	CONTINUE	SURT 960
136	RAD=0.1713E-8*(RBBTH(IG,J)**4-RBBL(IGH,J)**4)/	SURT 970
	.(REMH(IG,J)+REML(IGH,J)-1.0)	SURT 980
C	STORE RAD IN THE REML ARRAY.	SURT 990
	REML(IG,J)=RAD	SURT1000
	HX=CONH*(T(1,J)-T(I+1,J)-RAD*CONL*DELR(I)*RLN(I))/	SURT1010
	.(T(1,J)-T(I+1,J)+RAD*CONH)	SURT1020
	GO TO 138	SURT1030
140	HX=CONH	SURT1040
C	CALCULATE NEW BOUNDARY TEMPERATURES.	SURT1050
150	RBBH(IG,J)=(CONL*T(1+1,J)+HX/(DELR(I)*RLN(I))*T(I,J))/	SURT1060
	.(CONL+HX/(DELR(I)*RLN(I)))	SURT1070
	RBBL(IG,J)=RBBH(IG,J)	SURT1080
	GO TO 200	SURT1090
C	ARE TWO COOLANTS ADJACENT? -NO,YES,-	SURT1100
160	IF(GAPR(IG,J).GT.(-2.5E-10))GO TO 170	SURT1110
C	DO NOT ATTEMPT TO CALCULATE BOUNDARY TEMPERATURES.	SURT1120
	GO TO 200	SURT1130
C	CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE LOW	SURT1140
C	INDEX SIDE.	SURT1150
C	RECOVER DATA STORED IN THE REM-RBBT ARRAYS.	SURT1160
170	CONL=REMH(IG,J)	SURT1170
	CONH=RBBH(IG,J)	SURT1180
C	EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.	SURT1190
	IF(IG.EQ.1) GO TO 180	SURT1200
	IF(RATIO(IG,J).NE.200) GO TO 180	SURT1210
C	FIND THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT1220
	IGL=IG-1	SURT1230
	DO 174 II=1,IG1	SURT1240
	IG2=IG-II	SURT1250
	IF(GAPR(IG2,J).LT.(-.5E-10)) GO TO 172	SURT1260
	GO TO 174	SURT1270
C	EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT	SURT1280
C	THE LOW RADIAL BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT1290
172	IF(GAPR(IG2,J).LT.(-2.5E-10)) GO TO 180	SURT1300
	IGL=IG2	SURT1310
	GO TO 176	SURT1320
174	CONTINUE	SURT1330
C	RECOVER VALUE OF RAD STORED IN THE REML ARRAY.	SURT1340
176	RAD=REML(IGL,J)	SURT1350
	HX=CONH*(T(I+1,J)-T(I,J)+CONL*RAD*RATIOB(IG,J)*DELR(I)/RATIOK(IG))	SURT1360
	./(T(I+1,J)-T(I,J)-CONH*RAD*RATIOB(IG,J))	SURT1370

	GO TO 190	SURT1380
180	HX=CONH	SURT1390
C	CALCULATE NEW BOUNDARY TEMPERATURES.	SURT1400
190	RBBTL(I ₀ ,J)=(CONL*T(I,J)+HX*RATIOK(I ₀)/DELR(I)*T(I+1,J))/	SURT1410
	.(CONL+HX*RATIOK(I ₀)/DELR(I),	SURT1420
	RBBTL(I ₀ ,J)=RBBTL(I ₀ ,J)	SURT1430
C	=====	SURT1440
C	HAVE ALL POINTS ALONG THE CURRENT RADIAL-X GAPLINE BEEN	SURT1450
C	CONSIDERED ? -'YES,NO'-	SURT1460
200	CONTINUE	SURT1470
C	=====	SURT1480
C	HAVE ALL RADIAL-X BOUNDARY TEMPERATURES BEEN CALCULATED ?	SURT1490
C	-'YES,NO'-	SURT1500
210	CONTINUE	SURT1510
C	=====	SURT1520
C	=====	SURT1530
C	CALCULATE THE AXIAL-Y-THETA BOUNDARY TEMPERATURES.	SURT1540
	DO 410 J=1,JM	SURT1550
	JG=JGZ(J)	SURT1560
C	=====	SURT1570
C	EXCLUDE THE NON-GAPLINES.	SURT1580
	IF(JG.EQ.0)GO TO 410	SURT1590
C	=====	SURT1600
C	CONSIDER THE POINTS ALONG EACH AXIAL-Y-THETA GAPLINE.	SURT1610
	DO 400 I=2,IM	SURT1620
C	IS EITHER A GAP OR A COOLANT ADJACENT TO THE GAPLINE AT THIS	SURT1630
C	POINT ? -'NO,YES'-	SURT1640
	IF(GAPZ(JG,I).EQ.0.0)GO TO 400	SURT1650
C	IS IT A COOLANT OR A GAP WHICH IS ADJACENT ? -'COOLANT,GAP'-	SURT1660
	IF(GAPZ(JG,I).LT.0.0)GO TO 330	SURT1670
C	=====	SURT1680
C	CALCULATE THE BOUNDARY TEMPERATURES WHEN A GAP IS ADJACENT.	SURT1690
C	RECOVER DATA STORED IN THE ZEM-ZGBT ARRAYS.	SURT1700
	BX=ZB3L(JG,I)	SURT1710
	GK=ZB3H(JG,I)	SURT1720
	CONL=ZEM(JG,I)	SURT1730
	CONH=ZEMH(JG,I)	SURT1740
C	GET READY TO CALCULATE THE BOUNDARY TEMPERATURES OF THE GAP.	SURT1750
	B=ZATIO(JG,I)*CONL/CONH	SURT1760
	C=F(I,J)+B*F(I,J+1)	SURT1770
	RGAP=0.1713E-8*GAPZ(JG,I)/BX	SURT1780
C	PUT RGAP IN THE CORRECT UNITS FOR CIRCULAR GEOMETRY	SURT1790
	IF(ISHAPE.EQ.2) RGAP=RGAP*RP(I)	SURT1800
	ED=ZATIO(JG)*CONH/GAPZ(JG,I)	SURT1810
C	ALLOW 10 ITERATIONS TO CALCULATE THE BOUNDARY TEMPERATURES.	SURT1820
	DO 320 KK=1,10	SURT1830

IF(KK.EQ.1)GO TO 300	SURT1840
TH=ZBBTH(JG,I)	SURT1850
TB=ZBBTL(JG,I)	SURT1860
GO TO 310	SURT1870
300 TH=(1,J)	SURT1880
TB=(1,J+1)	SURT1890
310 GH=(GK+RGAP*(TH**2+TB**2)*(TH+TB))*EN	SURT1900
X1=G+G	SURT1910
X2=1,J+GH*(G+1.0)	SURT1920
ZBBTL(JG,I)=(X1+T(I,J+1))/X2	SURT1930
ZBBTH(JG,I)=(X1+T(I,J))/X2	SURT1940
C ARE THE BOUNDARY TEMPERATURES CONVERGED ? -'YES,NO'-	SURT1950
IF(ABS(ZBBTH(JG,I)-TH).LE.1.0)GO TO 400	SURT1960
C HAVE THE 10 ITERATIONS BEEN PERFORMED ? -'YES,NO'-	SURT1970
320 CONTINUE	SURT1980
C NEGATIVE TEMPERATURES ARE PRESENT AND RESULT IN LOCAL	SURT1990
C INSTABILITIES. TRY A SMALLER TIME STEP.	SURT2000
ERROR(1)=ISURT(1)	SURT2010
ERROR(2)=ISURT(3)	SURT2020
CALL ERROR2	SURT2030
C =====	SURT2040
C CALCULATE THE BOUNDARY TEMPERATURES WHEN A COOLANT IS ADJACENT.	SURT2050
C IS THE COOLANT ON THE HIGH INDEX SIDE OF THE GAPLINE ? -'NO,YES'-	SURT2060
330 IF(GAPZ(JG,I).LT.(-1.5E-10))GO TO 360	SURT2070
C CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE HIGH	SURT2080
C INDEX SIDE.	SURT2090
C SET GEOMETRICAL CONSTANT DXT FOR RECTANGULAR OR CYLINDRICAL	SURT2100
C GEOMETRY.	SURT2110
DXT=ZL(J)-ZP(J)	SURT2120
C CONVERT DXT FROM RADIANS TO FEET FOR CIRCULAR GEOMETRY.	SURT2130
IF(ISHAPE.EQ.2) DXT=DXT*RP(1)	SURT2140
C RECOVER DATA STORED IN THE ZEM-ZBBT ARRAYS.	SURT2150
CONLEZEML(JG,I)	SURT2160
CONH=ZBBTL(JG,I)	SURT2170
C EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.	SURT2180
IF(JG.EQ.JGMAX) GO TO 340	SURT2190
IF(MATG(JG,I).NE.200) GO TO 340	SURT2200
C FIND THE HIGH AXIAL-Y-THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK	SURT2210
JG1=JG+1	SURT2220
DO 334 JJ=JG1,JGMAX	SURT2230
IF(GAPZ(JJ,I).LT.(-1.5E-10)) GO TO 332	SURT2240
GO TO 334	SURT2250
C EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT	SURT2260
C THE HIGH AXIAL-Y-THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT2270
332 IF(GAPZ(JJ,I).LT.(-2.5E-10)) GO TO 340	SURT2280
JGH=JJ	SURT2290

	GO TO 33b	SURT2300
334	CONTINUE	SURT2310
336	RAD=0.1713E-8*((ZBBTH(JG,I)**4-ZBBTL(JGH,I)**4)/	SURT2320
	*(ZEMH(JG,I)+ZEML(JGH,I)-1.0)	SURT2330
C	STORE RAD IN THE ZEML ARRAY.	SURT2340
	ZEML(JG,I)=RAD	SURT2350
	HX=CONH*(T(I,J)-T(I,J+1)-RAD*CONL*DXT)/(T(I,J)-T(I,J+1)+RAD*CONH)	SURT2360
	GO TO 350	SURT2370
340	HX=CONH	SURT2380
C	CALCULATE NEW BOUNDARY TEMPERATURES.	SURT2390
350	ZBBTH(JG,I)=(CONL*T(I,J+1)+HX/DXT*T(I,J))/(CONL+HX/DXT)	SURT2400
	ZBBTL(JG,I)=ZBBTH(JG,I)	SURT2410
	GO TO 400	SURT2420
C	ARE TWO COOLANTS ADJACENT ? -NO,YES,-	SURT2430
360	IF(GAPZ(JG,I).GT.(-2.5E-10))GO TO 370	SURT2440
C	DO NOT ATTEMPT TO CALCULATE BOUNDARY TEMPERATURES.	SURT2450
	GO TO 400	SURT2460
C	CALCULATE THE BOUNDARY TEMPERATURES WITH A COOLANT ON THE LOW	SURT2470
C	INDEX SIDE.	SURT2480
C	SET GEOMETRICAL CONSTANT DXT FOR RECTANGULAR OR CYLINDRICAL	SURT2490
C	GEOMETRY.	SURT2500
370	DXT=2*P(J+1)-ZL(J)	SURT2510
C	CONVERT DXT FROM RADJANS TO FEET FOR CIRCULAR GEOMETRY.	SURT2520
	IF(ISHAPE.EQ.2) DXT=DXT*RP(I)	SURT2530
C	RECOVER THE DATA STORED IN THE ZEM-ZBBT ARRAYS.	SURT2540
	CONL=ZEMH(JG,I)	SURT2550
	CONH=ZBBTH(JG,I)	SURT2560
C	EXCLUDE CALCULATION OF RAD FOR COOLANTS WITH NO RADIATION.	SURT2570
	IF(JG.EQ.1) GO TO 380	SURT2580
	IF(MATZ(JG,I).NE.200) GO TO 380	SURT2590
C	FIND THE LOW AXIAL-Y-THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT2600
	JG1=JG-1	SURT2610
	DO 374 JJ=1,JG1	SURT2620
	JG2=JG-JJ	SURT2630
	IF(GAPZ(JG2,I).LT.(-.5E-10)) GO TO 372	SURT2640
	GO TO 374	SURT2650
C	EXCLUDE CALCULATION OF RAD WHERE TWO COOLANTS ARE ADJACENT AT	SURT2660
C	THE LOW AXIAL-Y-THETA BOUNDARY OF THE INTERNAL COOLANT BLOCK.	SURT2670
372	IF(GAPZ(JG2,I).LT.(-2.5E-10)) GO TO 380	SURT2680
	JG2=JG2	SURT2690
	GO TO 376	SURT2700
374	CONTINUE	SURT2710
C	RECOVER THE VALUE OF RAD STORED IN THE ZEML ARRAY.	SURT2720
376	RAD=ZEML(JG,I)	SURT2730
	HX=CONH*(T(I,J+1)-T(I,J)+CONL*RAD*DXT)/(T(I,J+1)-T(I,J)-CONH*RAD)	SURT2740
	GO TO 390	SURT2750

```

380 HX=CONL
C      CALCULATE NEW BOUNDARY TEMPERATURES.
390 ZBBTL(JG,I)=(CONL*T(I,J)+HX/DXT*T(I,I+1))/(CONL+HX/DXT)
    ZBBTH(JG,I)=ZBBTL(JG,I)
C      =====SURT2760
C      HAVE ALL POINTS ALONG THE CURRENT AXIAL-Y-THETA GAPLINE BEEN
C      CONSIDERED ? -'YES,NO'-
400 CONTINUE
C      =====SURT2770
C      HAVE ALL AXIAL-Y-THETA BOUNDARY TEMPERATURES BEEN CALCULATED ?
C      -'YES,NO'-
410 CONTINUE
C      =====SURT2780
    RETURN
    END
C      =====SURT2790
C      =====SURT2800
C      =====SURT2810
C      =====SURT2820
C      =====SURT2830
C      =====SURT2840
C      =====SURT2850
C      =====SURT2860
C      =====SURT2870
C      =====SURT2880
C      =====SURT2890
C      =====SURT2900

```

```

SUBROUTINE PRINT
INCLUDE COMDIM
C=
C PRINT THE TEMPERATURE DISTRIBUTION AND DATA REQUESTED BY THE USER
C VIA THE OPTIONS OR SUBROUTINE CUSTOM
C PUNCH THE LAST TEMPERATURE DISTRIBUTION
C=
C=
INTEGER TT
LOGICAL WRES
DIMENSION JTO (NQ),JTI (NQ),ITT (IQ),JTT1 (IQ),
2 JTT2 (IQ),
1 AT (IQ,JQ), CT (IQ,JQ),
3 JDT (IQ,JQ)
EQUIVALENCE (TT,AT),(TT,ITT),(TT,JTT1),
2 (TT(100),JTT2),
1 (TT,CT),(TT,JDT),(ITO,DTO),
2 (ITI,DTI)
DIMENSION NUMBER(60)
DATA NUMBER /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,
1 20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,
2 36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,
3 52,53,54,55,56,57,58,59,60/
C=
C=
DIMENSION FORMR (12),RPRINT(4),FORMAT (12),FORMZ (28),
2 FORMT1(28),IGAP (IG),JGAP (JG),ZPRINT(3),
4 FORMT2(28),WAY (4)
DATA FORMR /4H(13,, 4H12X,, 9*4H13X,, 1H)/
DATA RPRINT /6H16,10,, 6H17,16,, 6H18,4X,, 6H19,4X,/
DATA FORMZ /5H(140,, 6H13,2X,, 25*3H5X,, 1H)/
DATA ZPRINT /3H15,, 4H(4X,, 3H2X,/
DATA WAY /6H(1,J), 5H(J+1), 2H(I), 2H(J)/
C=
C=
IRET=0
WRES=.FALSE.
X1=CURTI*60.0
X2=CURTI*360.0
C SELECT THE TEMPERATURE DISTRIBUTION PRINT FORMAT
IF(VITER.EQ.LASTIP) 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
PRIN 10
PRIN 20
PRIN 30
PRIN 40
PRIN 50
PRIN 60
PRIN 70
PRIN 80
PRIN 90
PRIN 100
PRIN 110
PRIN 120
PRIN 130
PRIN 140
PRIN 150
PRIN 160
PRIN 170
PRIN 180
PRIN 190
PRIN 200
PRIN 210
PRIN 220
PRIN 230
PRIN 240
PRIN 250
PRIN 260
PRIN 270
PRIN 280
PRIN 290
PRIN 300
PRIN 310
PRIN 320
PRIN 330
PRIN 340
PRIN 350
PRIN 360
PRIN 370
PRIN 380
PRIN 390
PRIN 400
PRIN 410
PRIN 420
PRIN 430
PRIN 440
PRIN 450

```

```

ENTRY TEMPS                                PRIN 460
IRET=1                                      PRIN 470
167 WRITE(6,110) (ZA,I),I=1,12)           PRIN 480
110 FORMAT (1H1,30X,12A6,///)             PRIN 490
      IF(IRET.EQ.2) GO TO 169               PRIN 500
      WRITE (6,111)                         PRIN 510
111 FORMAT (49X,29HCOOLANT TEMPERATURES (F),/,3X,PRIN 520
114HCOOLANT NUMBER,4X,5HINLET,5X,6HOUTLET,4X,12HFLOW (LB/HR),8X,14HPRIN 530
2COOLANT NUMBER,4X,5HINLET,5X,6HOUTLET,4X,12HFLOW (LB/HR)) PRIN 540
C PREPARE ALL COOLANT DATA FOR PRINTING PRIN 550
DO 130 N=1,MAXFLO,2                        PRIN 560
  IF (NFLO(N).EQ.0) GO TO 150               PRIN 570
  N1=NFLO(N)                                PRIN 580
  N2=NFLO(N+1)                              PRIN 590
  ITI(N1)=TI(N1)-459.5                      PRIN 600
  IT0(N1)=T0(N1)-459.5                     PRIN 610
  IF(N1)=FLOW(N1)                           PRIN 620
  IF (N2.EQ.0 .OR. N.EQ.15) GO TO 120       PRIN 630
  ITI(N2)=TI(N2)-459.5                      PRIN 640
  IT0(N2)=T0(N2)-459.5                     PRIN 650
  IF(N2)=FLOW(N2)                           PRIN 660
  WRITE (6,140) N1,ITI(N1),IT0(N1),IF(N1),N2,ITI(N2),IT0(N2),IF(N2) PRIN 670
140 FORMAT(8X,I2,I16,I11,I14,15X,I2,I16,I11,I14) PRIN 680
      GO TO 130                             PRIN 690
120 WRITE (6,140) N1,ITI(N1),IT0(N1),IF(N1) PRIN 700
      GO TO 130                             PRIN 710
130 CONTINUE                               PRIN 720
150 IF(IRET.EQ.0) GO TO 153                 PRIN 730
169 WRITE(6,155) NITER                     PRIN 740
155 FORMAT(140,38X,I4,51H ITERATIONS PERFORMED WITHOUT REACHING STEADYPRIN 750
. STATE,/)                                PRIN 760
      IF(IRET.EQ.2) RETURN                   PRIN 770
      GO TO 157                             PRIN 780
153 WRITE(6,150) CURT I,X1,X2,NITER         PRIN 790
160 FORMAT (140,21H THE CURRENT TIME IS,F10.4,11H HOURS = ,F10.4,13PRIN 800
1H MINUTES = ,F13.5,8H SECONDS,4X,I11,31H ITERATIONS HAVE BEEN PERPRIN 810
2FORMED,/)                                PRIN 820
C= =====PRIN 830
C CONVERT TEMPERATURES TO INTEGRAL DEGREES FAHRENHEIT PRIN 840
157 DO 170 J=1,1MAX                        PRIN 850
      DO 170 I=1,JMAX                      PRIN 860
      TT(I,J)=T(I,J)-459.5                PRIN 870
170 CONTINUE                               PRIN 880
C SET THE MESH CORNERS IN THE PRINTOUT EQUAL TO ZERO PRIN 890
      TT(1,1)=0                           PRIN 900
      TT(1MAX,1)=0                        PRIN 910

```

```

      TT(1,JMAX)=0                                PRIN 920
      TT(1MAX,JMAX)=0                              PRIN 930
C= =====PRIN 940
C PRINT THE TEMPERATURE (INTEGRAL DEGRFES F)        PRIN 950
      WRITE (6,180)                                PRIN 960
180 FORMAT (1H0,47X,20HTEMPERATURES      (F))      PRIN 970
      CALL ARRAY (TT,DUM,1)                          PRIN 980
      IF(IRET.EQ.0) GO TO 159                        PRIN 990
      IRET=2                                          PRIN1000
      GO TO 107                                       PRIN1010
159 IF(IERROR(1).GT.0) WRITE(6,185) IERROR(1),NITER PRIN1020
185 FORMAT(1H0,5H*****,14,5H ERROR MESSAGES PRECEED THE RESULTS OF ITPRIN1030
      ,ERATION NO.,14,5H*****)                     PRIN1040
      LASTIP=NITER                                    PRIN1050
      IF (.NOT. DP) GO TO 280                         PRIN1060
C= =====PRIN1070
C PREPARE AND PRINT THE COOLANT AND NOHE TEMPERATURE DISTRIBUTION PRIN1080
C IN DECIMAL DEGREES FAHRENHEIT                     PRIN1090
190 WRITE (6,110) (ZA(I),I=1,12)                   PRIN1100
      WRITE (6,200)                                    PRIN1110
200 FORMAT (49X,28HCOOLANT TEMPERATURES      (F),/,1X,PRIN1120
114HCOOLANT NUMBER,5X,5HINLET,11X,6HOUTLET,7X,12HFLOW (LB/HR),4X,14PRIN1130
2HCOOLANT NUMBER,5X,5HINLET,11X,6HOUTLET,7X,12HFLOW (LB/HR)) PRIN1140
      DO 230 N=1,MAXFLO,2                             PRIN1150
      IF (NFLO(N).EQ.0) GO TO 250                     PRIN1160
      N1=NFLO(N)                                       PRIN1170
      N2=NFLO(N+1)                                     PRIN1180
      DTI(N1)=TI(N1)-460.0                             PRIN1190
      IF (DTI(N1).EQ. 0.0) DTI(N1)=0.0                PRIN1200
      DTO(N1)=TO(N1)-460.0                             PRIN1210
      IF (DTO(N1).EQ. 0.0) DTO(N1)=0.0                PRIN1220
      IF (N2.EQ.0 .OR. N.EQ.15) GO TO 220             PRIN1230
      DTI(N2)=TI(N2)-460.0                             PRIN1240
      IF (DTI(N2).EQ. 0.0) DTI(N2)=0.0                PRIN1250
      DTO(N2)=TO(N2)-460.0                             PRIN1260
      IF (DTO(N2).EQ. 0.0) DTO(N2)=0.0                PRIN1270
      WRITE (6,240) N1,DTI(N1),DTO(N1),FLOW(N1),N2,DTI(N2),DTO(N2),FLOW(N2) PRIN1280
142)                                                  PRIN1290
240 FORMAT (7X,12,7X,1PE13.7,3X,E13.7,3X,E13.7,1,2,7X,E13.7,3X,E13.7,3PRIN1300
1X,E13.7)                                             PRIN1310
      GO TO 250                                       PRIN1320
220 WRITE (6,240) N1,DTI(N1),DTO(N1),FLOW(N1)        PRIN1330
      GO TO 250                                       PRIN1340
230 CONTINUE                                         PRIN1350
250 WRITE (6,160) CURTI,X1,x2,NITER                 PRIN1360
C= =====PRIN1370

```

```

C      CONVERT THE TEMPERATURES TO DEGREES FAHRENHEIT
      DO 260 I=1,IMAX
      DO 260 J=1,JMAX
      DDT(I,J)=T(I,J)-460.0
      IF (DDT(I,J).EQ. 0.0) DDT(I,J)=0.0
260  CONTINUE
C      SET THE MESH CORNERS IN THE PRINTOUT EQUAL TO ZERO
      DDT(1,1)=0.0
      DDT(IMAX,1)=0.0
      DDT(1,JMAX)=0.0
      DDT(IMAX,JMAX)=0.0
C= =====
C      PRINT THE TEMPERATURES (DECIMAL DEGREES F)
      WRITE (6,270)
270  FORMAT (1H0,47X,28HDECIMAL TEMPERATURES      (F))
      CALL ARRAY (IDJM,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 (OP.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(6)) 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(1),PNAME(2)
290  FORMAT      (19X,32HSURFACE TEMPERATURES AT THE A6,PRIN1740
338H GAP OR COOLANT BOUNDARIES      (F),//,43X,4HTHE ,A6,62H(I) PRIN1750
43DIRECTION GAP OR COOLANT BOUNDARY GRIDLINES ARE HORIZONTAL/43X,4HTPRIN1760
5HE ,A6,33H(J) DIRECTION POINTS ARE VERTICAL) PRIN1770
C      CONVERT THE RADIAL-X SURFACE TEMPERATURES TO INTEGRAL DEGREES
C      FAHRENHEIT, AND INSURE THAT A POINT IN THE GAPLINE WHICH DOES
C      NOT BELONG TO A GAP OR COOLANT SURFACE APPEARS AS A BLANK IN
C      THE PRINT OUT
      V=0
      DO 291 I=1,IM

```

IF (IGR(1).EQ.0) GO TO 291	PRIN1640
N=N+1	PRIN1650
IGAP(N)=1	PRIN1660
291 CONTINUE	PRIN1670
NUM2=0	PRIN1680
292 NUM1=NUM2+1	PRIN1690
NUM2=NUM2+10	PRIN1900
IF (NUM2.GT.NRG) NUM2=NRG	PRIN1910
WRITE (6,293) (IGAP(L), L=NUM1,NUM2)	PRIN1920
293 FORMAT(1H0,I10,9I13)	PRIN1930
WRITE (6,294)	PRIN1940
294 FORMAT(1H0)	PRIN1950
DO 310 JJ=1,JMAX	PRIN1960
J=JJ	PRIN1970
IF (SW(21)) J=JMAX-JJ+1	PRIN1980
N=0	PRIN1990
NCOUNT=1	PRIN2000
DO 301 K=1,12	PRIN2010
301 FORMAT(K)=FORMR(K)	PRIN2020
DO 300 I=NUM1,NUM2	PRIN2030
NCOUNT=NCOUNT+1	PRIN2040
IT1=RBBTH(I,J)-459.5	PRIN2050
IF (IT1.LE.-450) GO TO 300	PRIN2060
IT2=RBBTL(I,J)-459.5	PRIN2070
IF (IT1.EQ.IT2 .AND. GAPR(I,J).LT.0.0) GO TO 302	PRIN2080
N=N+2	PRIN2090
ITT(N-1)=IT1	PRIN2100
ITT(N)=IT2	PRIN2110
FORMAT(NCOUNT)=RPRINT(2)	PRIN2120
IF (NCOUNT.EQ.2) FORMAT(NCOUNT)=RPRINT(1)	PRIN2130
GO TO 300	PRIN2140
302 N=N+1	PRIN2150
ITT(N)=IT1	PRIN2160
FORMAT(NCOUNT)=RPRINT(4)	PRIN2170
IF (NCOUNT.EQ.2) FORMAT(NCOUNT)=RPRINT(3)	PRIN2180
300 CONTINUE	PRIN2190
IF (N.EQ.0) GO TO 309	PRIN2200
WRITE (6,FORMAT) J,(ITT(I),I=1,N)	PRIN2210
GO TO 310	PRIN2220
309 WRITE (6,FORMAT) J	PRIN2230
310 CONTINUE	PRIN2240
IF (NUM2.LT.NRG) GO TO 292	PRIN2250
C=	PRIN2260
315 WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER	PRIN2270
WRITE (6,310) PNAME(2),PNAME(1),PNAME(2)	PRIN2280
316 FORMAT	PRIN2290


```

330H GAP OR COOLANT BOUNDARIES (F),//,43X,4HTHE ,A6,35H(I) PRIN2300
4DIRECTION POINTS ARE HORIZONTAL/43X,4HTHE ,A6,60H(J) DIRECTION GAPPRIN2310
5 OR COOLANT BOUNDARY GRIDLINES ARE VERTICAL) PRIN2320
C PRINT THE AXIAL-Y-THETA SURFACE TEMPERATURES (LOW AND HIGH) PRIN2330
C CONVERT THE AXIAL-Y-THETA SURFACE TEMPERATURES OF INTEGRAL DEGREESPRIN2340
C FAHRENHEIT, AND INSURE THAT A POINT IN THE GAPLINE WHICH DOES NOT PRIN2350
C BELONGS TO A GAP OR COOLANT SURFACE APPEARS AS A BLANK IN THE PRIN2360
C PRINT OUT PRIN2370
V=0 PRIN2380
DO 320 J=1,J1 PRIN2390
IF (JGZ(J).EQ.0) GO TO 320 PRIN2400
V=V+1 PRIN2410
JGAP(V)=J PRIN2420
320 CONTINUE PRIN2430
NUM2=0 PRIN2440
321 NUM1=NUM2+1 PRIN2450
NUM2=NUM2+25 PRIN2460
IF (NUM2.GT.1MAX) NUM2=1MAX PRIN2470
WRITE (6,322) (NUMBER(L),L=NUM1,NUM2, PRIN2480
322 FORMAT(1H0,5X,25I5) PRIN2490
DO 330 JJ=1,NZG PRIN2500
J=JJ PRIN2510
IF (JW(Z1)) J=NZG-JJ+1 PRIN2520
V1=0 PRIN2530
V2=0 PRIN2540
NCOUVT=2 PRIN2550
DO 331 K=1,25 PRIN2560
FORM11(K)=FORMZ(K) PRIN2570
FORM22(K)=FORMZ(K) PRIN2580
331 CONTINUE PRIN2590
DO 335 I=NUM1,NUM2 PRIN2600
NCOUVT=NCOUVT+1 PRIN2610
JT1=73314(J,1)-459.5 PRIN2620
IF (JT1.EQ.-450) GO TO 335 PRIN2630
V1=V1+1 PRIN2640
JTT1(V1)=JT1 PRIN2650
FORM11(NCOUVT)=7PRINT(1) PRIN2660
JT2=73314(J,1)-459.5 PRIN2670
IF (JT1.EQ.JT2 .AND. GAPZ(J,I).LT.0.0) GO TO 335 PRIN2680
V2=V2+1 PRIN2690
JTT2(V2)=JT2 PRIN2700
FORM22(NCOUVT)=7PRINT(1) PRIN2710
335 CONTINUE PRIN2720
IF (V1.EQ.0) GO TO 326 PRIN2730
WRITE (6,FORMAT1) JGAP(J),(JTT1(I),I=1,V1) PRIN2740
IF (V2.EQ.0) GO TO 328 PRIN2750

```

```

      FORM12(1)=ZPRINT(2)                                PRIN2760
      FORM12(2)=ZPRINT(3)                                PRIN2770
      WRITE (6,FORM12) (JTT2(1),I=1,N2)                 PRIN2780
      GO TO 300                                           PRIN2790
326 WRITE (6,FORM11) JGAP(J)                             PRIN2800
328 WRITE (6,295)                                         PRIN2810
295 FORM11 (1H)                                           PRIN2820
330 CONTINUE                                             PRIN2830
      IF (NUM2.LT.IMAX) GO TO 321                         PRIN2840
C= =====PRIN2850
C SHOULD THE HEAT RATES AND HEAT FLUXES BE PRINTED? -'NO,YES'- PRIN2860
340 IF (.NOT.SW(5)) GO TO 500                             PRIN2870
C= =====PRIN2880
C CALCULATE THE RADIAL-X HEAT RATES                     PRIN2890
      DO 350 J=1,JMAX                                     PRIN2900
      DO 350 I=1,IMAX                                     PRIN2910
      AT(1,J)=RR(I,J)*K(R(I,J)*(T(I,J)-T(I+1,J)))       PRIN2920
350 CONTINUE                                             PRIN2930
C PRINT THE RADIAL-X HEAT RATES                         PRIN2940
355 WRITE(6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER      PRIN2950
360 FORMAT (1H1,30X,12A6,/,21H THE CURRENT TIME IS,F10.4,11H HOURS PRIN2960
      =,F10.4,13H MINUTES =,F13.5,8H SECONDS,4X,14,31H ITERATIONS HPRIN2970
      2AVE BEEN PERFORMED,///)
      IF (NRES) GO TO 275                                PRIN2980
      WRITE (6,370) PNAME(1),WAY(1)                     PRIN3000
370 FORMAT (19X,15HHEAT RATE IN ,A6,43H DIRECTION BETWEEN POINPRIN3010
      INTS (I,J) AND (,A6,5X,8H(BTU/HR))
      CALL ARRAY (IDUM,AT,2)                             PRIN3030
C= =====PRIN3040
C CALCULATE THE RADIAL-X HEAT FLUXES                     PRIN3050
      NAME=ISHAPE+1                                       PRIN3060
      DO 410 J=1,JMAX                                     PRIN3070
      DO 410 I=1,IMAX                                     PRIN3080
      GO TO (380,390,400), NAME                           PRIN3090
380 AT(1,J)=AT(1,J)/(RR(I,J)*RL(I)*ALOG(OP(I+1)/RP(I))) PRIN3100
      GO TO 410                                           PRIN3110
390 AT(1,J)=AT(1,J)/(RR(I,J)*DELR(I))                   PRIN3120
      GO TO 410                                           PRIN3130
400 AT(1,J)=AT(1,J)/(RR(I,J)*ALOG(RP(I+1)/RP(I))*RL(I)) PRIN3140
410 CONTINUE                                             PRIN3150
C PRINT THE RADIAL-X HEAT FLUXES                         PRIN3160
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER    PRIN3170
      WRITE (6,420) PNAME(1),WAY(3),WAY(1)              PRIN3180
420 FORMAT (19X,16HHEAT FLUX IN ,A6,49H DIRECTION BASED ON PRIN3190
      1THE AREA OF GRIDLINE (,A2,/,28X,30HBETWEEN POINTS (I,J) AND PRIN3200
      2 (,A6,5X,14H(BTU/HR-FT**2))                      PRIN3210

```

```

      CALL ARRAY (IDJW,AT,2)
C= =====PRIN3220
C CALCULATE THE AXIAL-Y-THETA HEAT RATE
      DO 430 J=1,JMAX
      DO 430 I=1,IMAX
      AT(I,J)=RZ(I,J)*AZ(I,J)*(T(I,J)-T(I,J+1))
      430 CONTINUE
C PRINT THE AXIAL-Y-THETA HEAT RATE
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,370) PNAME(2),WAY(2)
      CALL ARRAY (IDJW,AT,2)
C= =====PRIN3230
C CALCULATE THE AXIAL-Y-THETA HEAT FLUXES
      DO 460 J=1,JM
      DO 460 I=1,IMAX
      GO TO (450,460,470), NAME
      450 AT(I,J)=AT(I,J)/(RZ(I,J)*(ZP(J+1)-ZP(J)))
      GO TO 460
      460 AT(I,J)=AT(I,J)/(RZ(I,J)*DELZ(J))
      GO TO 460
      470 AT(I,J)=AT(I,J)/(RZ(I,J)*(RP(I)*(ZP(J+1)-ZP(J))))
      480 CONTINUE
C PRINT THE AXIAL-Y-THETA HEAT FLUXES
      WRITE (6,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,420) PNAME(2),WAY(4),WAY(2)
      CALL ARRAY (IDJW,AT,2)
C= =====PRIN3330
C SHOULD THE EFFECTIVE THERMAL CONDUCTIVITIES AND THERMAL
      -CONDUCTANCES BE PRINTED?,-'NO,YES'-
      500 IF (.NOT.SW(7)) GO TO 610
C= =====PRIN3340
C PRINT THE EFFECTIVE RADIAL-X THERMAL CONDUCTIVITIES
      DO 510 I=1,IMAX
      KR(I,1)=0.0
      510 KR(I,JMAX)=0.0
      DO 520 J=1,JMAX
      KR(I,MAX,J)=0.0
      WRITE (6,500) (ZA(I),I=1,12),CURTI,X1,X2,NITER
      WRITE (6,530) PNAME(1),WAY(1)
      530 FORMAT (19X,11HEFFECTIVE ,A6,404 CONDUCTIVITY BETWEEN POINPRIN3610
      ITS (I,J) AND (,A6,5X,13H(STU/HR-FT-F))
      CALL ARRAY (IDJW,KR,2)
C= =====PRIN3640
C PRINT THE EFFECTIVE AXIAL-Y-THETA THERMAL CONDUCTIVITIES
      DO 540 I=1,IMAX
      540 KZ(I,JMAX)=0.0
      PRIN3650
      PRIN3660
      PRIN3670

```

```

DO 550 J=1,JMAX
KZ(I,J)=0.0
550 KZ(IMAX,J)=0.0
WRITE (0,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
WRITE (0,530) PNAME(2),NAY(2)
CALL ARRAY (IDJM,KZ,2)
C=
=====PRIN3740
C PRINT THE RADIAL-X THERMAL CONDUCTANCES
PRIN3750
DO 570 I=1,IMAX
PRIN3760
DO 570 J=1,JMAX
PRIN3770
CT(I,J)=KR(I,J)*KR(I,J)
PRIN3780
570 CONTINUE
PRIN3790
WRITE (0,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
PRIN3800
WRITE (0,580) PNAME(1),NAY(1)
PRIN3810
580 FORMAT (19X,A6,54H THERMAL CONDUCTANCE BETWEEN POINTS (I,PRIN3820
1J) AND (A6,5X,10H(BTU/HR-F))
PRIN3830
CALL ARRAY (IDJM,CT,2)
PRIN3840
C=
=====PRIN3850
C PRINT THE AXIAL-Y-THETA THERMAL CONDUCTANCES
PRIN3860
DO 590 I=1,IMAX
PRIN3870
DO 590 J=1,JMAX
PRIN3880
CT(I,J)=KZ(I,J)*KZ(I,J)
PRIN3890
590 CONTINUE
PRIN3900
WRITE (0,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
PRIN3910
WRITE (0,580) PNAME(2),NAY(2)
PRIN3920
CALL ARRAY (IDJM,CT,2)
PRIN3930
C=
=====PRIN3940
C SHOULD THE OUTPUT BE PUT ON A TAPE? _'NO,YES'-
PRIN3950
610 IF (.NOT.SW(3)) GO TO 620
PRIN3960
C=
=====PRIN3970
C PUT THE CURRENT TEMPERATURE DISTRIBUTION ON TAPE
PRIN3980
WRITE (OUTTAP) NITER,CURTI,((RBT(I,J),RBT(I,J),IG=1,NRG),J=1,PRIN3990
1,JMAX),((ZBT(I,J),ZBT(I,J),JG=1,NZG),I=1,IMAX), (FLOW(M),TI(MPRIN4000
2),IJ(M),M=1,MAXFLO),((T(I,J),I=1,IMAX),J=1,JMAX)
PRIN4010
C=
=====PRIN4020
620 IF (.NOT.OP) GO TO 640
PRIN4030
WRITE (0,360) (ZA(I),I=1,12),CURTI,X1,X2,NITER
PRIN4040
WRITE (0,1000)
PRIN4050
1000 FORMAT (52X,15HHEAT BALANCE,///,29X,15HHEAT GENERATED,25X,
PRIN4060
A28HHEAT LOST/GAINED BY COOLANTS, //, 25X,12HLOCK NUMBER,9X,
PRIN4070
A6H3J/HR,13X,14HCOOLANT NUMBER,8X,6HQTU/HR,/)
PRIN4080
SUMM=0.0
PRIN4090
SUML=0.0
PRIN4100
DO 1200 L=1,LMAX
PRIN4110
ILS=IL(L)
PRIN4120
IHS=IH(L)
PRIN4130

```

JLS=JL(L)	PRIN4140
JHS=JH(L)	PRIN4150
SUM=0.0	PRIN4160
C TEST WHETHER A BLOCK IS A COOLANT OR NOT	PRIN4170
IF (M3(L) .GT. 0) GO TO 1100	PRIN4180
N=IM3(M3(L))	PRIN4190
II=IABS(IPATH(N))	PRIN4200
GO TO (1025,1050,1050), II	PRIN4210
C AXIAL COOLANT FLOW	PRIN4220
1050 DO 1060 J=JLS,JHS	PRIN4230
IF (ILS .LE. 2) GO TO 1055	PRIN4240
IL=ILS-1	PRIN4250
SUM=SUM + RR(I,J) * KR(I,J) * (T(I,J)-T(ILS,J))	PRIN4260
1055 IF (IHS .GE. IM) GO TO 1060	PRIN4270
SUM = SUM - RR(IHS,J) * KR(IHS,J) * (T(IHS,J) - T(IHS+1,J))	PRIN4280
1060 CONTINUE	PRIN4290
IF (II.NE. 3) GO TO 1075	PRIN4300
C RADIAL COOLANT FLOW	PRIN4310
1025 DO 1040 I=ILS,IHS	PRIN4320
IF (JLS .LE. 2) GO TO 1030	PRIN4330
J=JLS-1	PRIN4340
SUM=SUM + RZ(I,J) * KZ(I,J) * (T(I,J) - T(I,JLS))	PRIN4350
1030 IF (JHS .GE. JM) GO TO 1040	PRIN4360
SUM=SUM - RZ(I,JHS) * KZ(I,JHS) * (T(I,JHS) - T(I,JHS+1))	PRIN4370
1040 CONTINUE	PRIN4380
1075 SUMC=SUMC + SUM	PRIN4390
WRITE (6,1050) N,SUM	PRIN4400
1080 FORMAT (69X,I4,9X,1PE14.7)	PRIN4410
GO TO 1200	PRIN4420
1100 DO 1160 I=ILS,IHS	PRIN4430
DO 1160 J=JLS,JHS	PRIN4440
SUM = SUM + W(I,J)	PRIN4450
1160 CONTINUE	PRIN4460
SUMM = SUMM + SUM	PRIN4470
IF (SUM .EQ. 0.0) GO TO 1200	PRIN4480
WRITE (6,1170) M3(L),SUM	PRIN4490
1170 FORMAT (28X,I4,9X,1PE14.7)	PRIN4500
1200 CONTINUE	PRIN4510
WRITE (6,1205) SUMM,SUMC	PRIN4520
1205 FORMAT (1H0,39X,16H-----,25X,16H-----,/,	PRIN4530
x41X,1PE14.7,27X,1PE14.7)	PRIN4540
C =====	PRIN4550
C RESTORE THE SECOND TEMPERATURE ARRAY	PRIN4560
640 DO 650 J=1,JMAX	PRIN4570
DO 650 I=1,IMAX	PRIN4580
AT(I,J)=T(I,J)	PRIN4590
650 CONTINUE	PRIN4600
C= =====	PRIN4610
RETURN	PRIN4620
END	PRIN4630

```

SUBROUTINE CUSTOM                                CUST 10
INCLUDE COMDIM                                CUST 20
C= =====CUST 30
C DO WHAT THE USER WANTS TO DO                CUST 40
C= =====CUST 50
C= =====CUST 60
C= =====CUST 70
C= =====CUST 80
C= =====CUST 90
C= =====CUST 100
C= =====CUST 110
C= =====CUST 120
RETURN                                CUST 130
END                                CUST 140

```

```

SUBROUTINE PUN                                PUN 10
INCLUDE COMDIM                                PUN 20
C =====PUN 30
C PUNCH THE FINAL TEMPERATURE DISTRIBUTION.    PUN 40
C =====PUN 50
C =====PUN 60
DIMENSION ALABEL(4), TPUN(1)                PUN 70
EQUIVALENCE (TT,TPUN)                        PUN 80
DATA ALABEL /DHHEADER, DHTEMP, DHINLET, DHOUTLT/ PUN 90
C =====PUN 100
C =====PUN 110
WRITE (6,210) (ZA(I),I=1,12)                PUN 120
IF(.NOT.SW(10)) GO TO 300                    PUN 130
CURTI=0.0                                    PUN 140
NITER=0                                       PUN 150
300 BMAX=IMAX                                 PUN 160
    BJMAX=JMAX                                PUN 170
    BNITER=NITER                              PUN 180
    WRITE (6,220) CURTI,BMAX,BJMAX,BNITER,ALABEL(1) PUN 190
    PUNCH 250, CURTI,BMAX,BJMAX,BNITER,ALABEL(1) PUN 200
    NT=0                                       PUN 210
    DO 240 J=1,JMAX                           PUN 220
    DO 240 I=1,IMAX                           PUN 230
    NT=NT+1                                    PUN 240
    TPUN(NT)=T(I,J)                          PUN 250
240 CONTINUE                                PUN 260
    CALL PUNCHY (TPUN,NT,ALABEL(2))           PUN 270
    CALL PUNCHY (TI,MAXFLO,ALABEL(3))         PUN 280
    CALL PUNCHY (TO,MAXFLO,ALABEL(4))         PUN 290
    PUNCH 250                                  PUN 300
    WRITE (6,260)                              PUN 310
    RETURN                                     PUN 320
C= =====PUN 330
210 FORMAT (141,30X,12A6,///,40X,34HPUNCHED TEMPERATURE DECK (R)PUN 340
1,/)                                          PUN 350
220 FORMAT (10X,12.0,3F12.1,24X,A6)         PUN 360
230 FORMAT (12.0,3F12.1,24X,A6)             PUN 370
250 FORMAT (////)                            PUN 380
260 FORMAT (140,20X,52H(REMOVE THE FIVE BLANK CARDS AT END OF PUNCHED PUN 390
1DECK))                                     PUN 400
C= =====PUN 410
CE *****PUN 420
CE *****PUN 430
CE *****PUN 440
CE *****PUN 450
SUBROUTINE PUNCHY (TP,LENGTH,ANAME)          PUN

```

```

C= =====PUN 460
C PUNCH THE TEMPERATURE DISTRIBUTION DECK PUN 470
C=====PUN 480
C=====PUN 490
DIMENSION TP ( 1),FORM ( 9),FORMAT ( 9),FB ( 3) PUN 500
DATA FORM /6H , 6H 1P, 6*6HE12.6,, 6HA5,I3)/ PUN 510
DATA FB /6H( 10X,, 6H( , 6H 12X,/ PUN 520
C=====PUN 530
N=LENGTH/6 PUN 540
NLEFT=LENGTH-N*6 PUN 550
NN1=1 PUN 560
NN2=6 PUN 570
DO 90 I=1,9 PUN 580
FORM(I)=FORMAT(I) PUN 590
90 CONTINUE PUN 600
DO 100 KK=1,N PUN 610
FORM(1)=FB(1) PUN 620
WRITE (6,FORM) (TP(I),I=NN1,NN2),ANAME,KK PUN 630
FORM(1)=FB(2) PUN 640
PUNCH FORM, (TP(I),I=NN1,NN2),ANAME,KK PUN 650
NN1=NN2+1 PUN 660
NN2=NN2+6 PUN 670
100 CONTINUE PUN 680
C=====PUN 690
IF (NLEFT.EQ.0) GO TO 120 PUN 700
KK=4+1 PUN 710
J=NLEFT+3 PUN 720
DO 110 I=J,8 PUN 730
FORM(1)=FB(3) PUN 740
110 CONTINUE PUN 750
FORM(1)=FB(1) PUN 760
WRITE (6,FORM) (TP(I),I=NN1,LENGTH),ANAME,KK PUN 770
FORM(1)=FB(2) PUN 780
PUNCH FORM, (TP(I),I=NN1,LENGTH),ANAME,KK PUN 790
C=====PUN 800
120 RETURN PUN 810
END PUN 820

```


1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

REFERENCES

1. 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.
2. 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
3. Clark, S. S., Del Bene, J. V. and Petersen, J. F., "TAC3D, A General Purpose Three-Dimensional Heat Transfer Computer Code - Mathematical Formulations and Programmer's Guide," USAEC Report GA-9264, Gulf General Atomic, September 1969.
4. Peaceman, D. W. and Rachford, H. H., "The Numerical Solution of Parabolic and Elliptic Differential Equations," I. Soc. Indust. Appl. Math1, Vol. 3, No. 1, pp. 28-41, March, 1955.
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. 227, 1959.
7. Jahnke, E. and Emde, F., Tables of Functions, Dover Publications, 4th Edition, 1945.