Gulf General Atomic Incorporated

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

AEC RESEARCH AND
DEVELOPMENT REPORT

GA-9263

TAC3D

A GENERAL PURPOSE THREE-DIMENSIONAL
HEAT TRANSFER COMPUTER CODE

- USER'S MANUAL -

Work done by:

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

Report written by:

J. F. Petersen

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

LEGAL NOTICE

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

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

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

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

Gulf General Atomic Project 317.1801

September, 1969

DISTRIBUTION OF THIS REPORT IS UNLIMITED

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.

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

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

ABSTRACT

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

Some advantages of the code are:

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

The principal limitations of the code are:

1. The grid plane system must be orthogonal in the rectangular or cylindrical coordinate system. Therefore, the sides of the nodal elements must also be orthogonal. The entire problem must be bounded by six grid planes in one of the coordinate systems. Difficulties in treating irregular boundaries can be overcome to

some extent through the use of materials having specially chosen properties.

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

TAC3D has been assigned operational status. The machine requirement is a 65K Univac 1108, or equivalent. In addition to input-output, a maximum of four and a minimum of no tapes 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 of the problem and is not easily defined.

A related code, TAC2D, is two-dimensional and has all the features of TAC3D described in this abstract. All symbolic elements, 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 . . ,	viii
1. INTRODUCTION	1
2. INPUT DEFINITION	5
2.1 GEOMETRY AND THERMAL PARAMETERS	5
2.1.1 Grid Planes and Points	5
2.1.2 Thermal Parameters	7
2.1.3 Boundaries and Blocks	8
2.1.4 Coolant Blocks	8
2.1.5 Gaps	10
2.1.6 Gap Planes	11
2.2 INITIAL TEMPERATURES	11
2.2.1 Specified Initial Temperatures	11
2.2.2 Previously Punched Temperatures	12
2.3 THE TIME VARIABLE	12
2.3.1 Initial Time Value	12
2.3.2 Time Incrementation	12
2.3.3 Initial Iteration	13
2.4 THE STEADY-STATE OPTION	13
2.5 THERMAL PARAMETER FUNCTIONS	15
2.5.1 Function Names	15
2.5.2 Function Variables	17
2.5.3 Standard Function Values	20
2.5.4 Interconnection of Coolants	20
2.5.5 Function Control Constants	21
2.5.6 Additional Arithmetic Statement Functions	21
2.5.7 Function Subprograms	22

Table of Contents (Continued)

3. OUTPUT AND OUTPUT CONTROL	23
3.1 GENERAL	23
3.2 BASIC OUTPUT	23
3.3 AUXILIARY OUTPUT	24
3.3.1 Overlay	24
3.3.2 Overall Heat Balance	25
3.3.3 Residuals	25
3.4 OPTIONAL OUTPUT	25
3.4.1 Decimal Temperatures	25
3.4.2 Geometry Factors, Effective Conductivities and Effective Conductances	26
3.4.3 Heat Rates and Heat Fluxes	26
3.4.4 Surface Temperatures	27
3.4.5 Punched Output	27
3.4.6 Tape Output	28
3.5 SPECIAL OUTPUT	28
4. INPUT FORMULATION	29
4.1 DATA GROUPS	29
4.1.1 Thermal Parameter Functions	29
4.1.2 Titles	30
4.1.3 Indicators	31
4.1.4 X Grid Planes	33
4.1.5 Y Grid Planes	33
4.1.6 Z Grid Planes	34
4.1.7 Block Information	34
4.1.8 Specified Initial Temperatures	40
4.1.9 Coolant Limits	41
4.1.10 Time History	44
4.1.11 Function Control Constants	46
4.1.12 Previously Punched Temperatures	46
4.2 CARD INPUT FOR A COMPUTER RUN	48
4.3 PERFORMANCE OF A COMPUTER RUN	51
4.3.1 Execution Time and Printed Page Limits for Use with the Univac 1108 Computer	51
4.3.2 Error Messages	52

Table of Contents (Continued)

5.	TRUNCATION AND ROUND OFF ERRORS	53
5.1	STABILITY OF SOLUTION	53
5.1.1	Basic Considerations	53
5.1.2	Practical Approaches	54
5.2	FINITE DIFFERENCE FORMULATION	56
5.2.1	All Results	56
5.2.1.1	Grid Spacing	56
5.2.2	Transient Results	56
5.2.2.1	Nonlinearity of Time	56
5.2.2.2	Evaluation of Time and Temperature Dependent Quantities	56
5.2.2.3	Solution for Coolant Temperatures	57
5.3	ONE-DIMENSIONAL TREATMENT OF RADIATION	59
5.4	ROUND OFF ERRORS	60
6.	TECHNIQUES	61
6.1	BOUNDARY CONDITIONS	61
6.1.1	Adiabatic Boundary	61
6.1.2	Constant Temperature Boundary	62
6.1.3	Boundary with Radiation	62
6.2	IRREGULAR EXTERNAL BOUNDARIES	63
6.3	INTERNAL DUMMY MATERIALS	66
	APPENDIX A - PROGRAM ERROR MESSAGES	67
	APPENDIX B - EXAMPLE USE OF ADDITIONAL STATEMENT FUNCTIONS	71
	APPENDIX C - EXAMPLE USE OF FUNCTION SUBPROGRAMS	73
	APPENDIX D - DIMENSIONAL LIMITS	75
	APPENDIX E - DUMPING	77
E.1	FULL CORE DUMP	77
E.2	DUMP OF A GIVEN COMMON ELEMENT	77

Table of Contents (Continued)

APPENDIX F - OUTPUT DATA TAPE	79
REFERENCES	81

LIST OF FIGURES

Fig. 1. Coordinate systems in TAC3D	2
Fig. 2. Typical Z plane of a three-dimensional problem	6
Fig. 3. Coolant blocks in TAC3D	9
Fig. 4. Gap and coolant limitations illustrated in a cross- section normal to the Z axis	39
Fig. 5. Irregular external boundaries, Example 1	64
Fig. 6. Irregular external boundaries, Example 2	65

1. INTRODUCTION

The digital computer code TAC3D* was developed at Gulf General Atomic for obtaining temperature solutions in the wide variety of three-dimensional thermal systems which are encountered in the field of nuclear engineering. Code calculations are governed by the heat conduction equation:

$$\nabla \cdot k \nabla T + q''' = \frac{\partial}{\partial t} \rho c T$$

where

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

T is local temperature, °F

q''' is volumetric heat generation rate, Btu/hr-ft³

ρ is density, lb/ft³

c is specific heat, Btu/lb-°F

t is time, hr

This equation is replaced by an equivalent set of linear finite difference equations, which is solved for the local temperatures at given points in time by the implicit numerical method given in Ref. 1. Steady-state results are found by extending a transient calculation to the limit where thermal equilibrium is attained. An option is available for performing this pseudo-transient calculation as efficiently as possible. If it is used, specific heats and time increments are determined as a part of the calculation rather than being given as input. In the finite difference equations, the local value of k may be an effective overall thermal conductivity which includes the effects of convection and/or radiation.

The problem must be modeled within the geometry envelopes of one of the two coordinate systems shown in Fig. 1. The choice depends upon whether

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

Fig. 1. Coordinate systems in TAC3D

it is best described as a rectangular parallelepiped or an incomplete* right circular cylinder. The code includes provision for both internal and external coolants. Use of internal coolants is optional, but coolants flowing on the six outer surfaces must always be included to describe boundary conditions by assigning appropriate values to the coolant thermal parameters. There is provision for internal thermal radiation, but its treatment is one-dimensional.

The purpose of this document is to provide all information required for the use of TAC3D. The mathematical formulations used, a programmer's description of the code, and a listing are given in Ref. 2.

TAC3D is actually one of two generalized heat transfer codes which have been developed at Gulf General Atomic. The other code is TAC2D, which is a two-dimensional version of TAC3D and is described in Refs. 3 and 4. Although the following background remarks are written with reference to TAC3D, they apply as well to TAC2D.

A code was needed which could be easily used by persons not familiar with computer science. Toward this end, care was taken to keep all input and output within the scope of engineering terminology. Also, a system of input checking and easily interpreted error messages was included. As a final step, this user's manual has been prepared to provide a comprehensive guide to code application and input. The result of the above provisions is that TAC3D is a "black box" type of code in which the user should be detached from programming and computer system aspects of the problem solution.

The features most desired in codes such as TAC3D are generality, simplicity of input, and economy of computer time. Generality can usually be increased only by partially sacrificing the latter two features. TAC3D was formulated under the basic philosophy of obtaining a trade-off among all three features which would be an optimum for economical solution of thermal problems typically encountered in the nuclear field.

*The cylinder is termed incomplete because, although the coordinate θ , illustrated in Fig. 1, may extend from 0° through 360° , there is no connection between the planes where $\theta = 0^\circ$ and $\theta = 360^\circ$.

General purpose heat transfer codes are usually developed in terms of a network of points connected by thermal resistances. In most codes, the arrangement of these points may be purely arbitrary. A high degree of generality is obtained at the expense of input simplicity, since an individual set of data must usually be supplied for each point. If, on the other hand, it is chosen to confine the problem within the geometry envelope of one of the two coordinate systems shown in Fig. 1, the input complexity may be greatly reduced. The entire geometry and subdivision can be defined by giving the coordinates of constant coordinate planes, or grid planes, such as those illustrated in the figure. If the points are defined as lying midway between adjacent grid planes, then a region of points can be established by giving the six bounding grid planes of that region. Other input data, such as that required to specify the resistances, can be given in condensed form by referring to these regions rather than to the individual points which they contain. The above approach was used because experience had indicated that many of the three-dimensional thermal systems for which temperature calculations were being performed at Gulf General Atomic could be modeled within one of the geometry envelopes shown in Fig. 1. Furthermore, the computational algorithm applied in the solution for the point temperatures is the most efficient known and could not have been used had complete generality been maintained in the arrangement of the points.

For simplicity of illustration, the basic presentation of the definitions and instructions which follow is in Cartesian rather than cylindrical coordinates. Within this report, the correspondences

$$X = R \text{ (radial), } Y = Z \text{ (axial), } Z = \theta \text{ (theta)}$$

are used for translation between the two coordinate systems.

2. INPUT DEFINITION

2.1 GEOMETRY AND THERMAL PARAMETERS

This section describes how the system of calculation points is established and how the sets of thermal parameters governing the behavior of these points are identified and located.

2.1.1 Grid Planes and Points

In any of the two coordinate systems shown in Fig. 1, there are three sets of grid planes which are defined as surfaces along which the value of one of the three coordinates remains constant. Grid planes are identified by the direction of a vector normal to the plane. For example, a grid plane whose normal vector is parallel to the Z axis is called a Z grid plane, or simply a Z plane. In the cylindrical coordinate system, the radial grid planes are really not planes but instead are cylindrical surfaces. The notation "plane" is still maintained. Grid planes are the primary means by which the problem is subdivided and are specified as input by giving their constant coordinate values. The two extreme value grid planes of each set are the external boundaries of the problem. The lower value of each set may, but need not, be equal to zero. No grid plane may have a negative value.

The relationship of the calculation points to the grid planes is illustrated in Fig. 2. The points are of two types, internal and external. An internal point is located midway between the adjacent grid planes of each coordinate direction. The temperature calculated for the point will be the temperature at the location given by the point coordinates. An external point is located outside of an external boundary midway between the two sets of adjacent grid planes which intersect that boundary. The temperature calculated for an external point is the temperature of the external source or sink directly adjacent to the point. The temperatures of external points reflect local boundary conditions only.

LEGEND:

- GRID PLANE
- BLOCK BOUNDARY OR EXTERNAL BOUNDARY
- DENOTES A GAP ON A BOUNDARY
- DENOTES A COOLANT FLOWING IN THE INDICATED DIRECTION
- INTERNAL POINT
- ▲ EXTERNAL BOUNDARY POINT

Fig. 2. Typical Z plane of a three-dimensional problem

2.1.2 Thermal Parameters

Every part of a problem is defined by one of the following sets of thermal parameters. These sets are labeled materials, coolants, and gases. Their assigned thermal parameters are:

Materials

Thermal conductivities in the X, Y, and Z directions

Volumetric specific heat

Emissivities in the +X, -X, +Y, -Y, +Z, or -Z directions

Volumetric heat generation rate

Coolants

Specific heat capacity (constant pressure)

Reynolds number

Heat transfer coefficient

Inlet temperature

Mass flow rate

Gases

Thermal conductivity

Coolants are flowing coolants, in that the temperatures of points in coolants are influenced by an inlet temperature, a flow rate, and a specific heat. Gases (in gaps) are stagnant. Their presence serves only to modify the thermal resistances between points which are adjacent in solids. As is discussed in 2.1.5, gases contain no points.

There is provision in the program for fifteen different members in each set. They are identified by number. Each of the materials, coolants, and gases used is assigned a number 1 through 15. Not all of the member numbers must be used, but two members of the same set must always be assigned different numbers whenever any of their thermal parameters are not identical.

The actual values of the thermal parameters are given by FORTRAN V arithmetic statement functions as described in Section 2.5.

2.1.3 Boundaries and Blocks

The highest and lowest index value grid planes in each coordinate direction (a total of six) are always the external boundaries of the problem, as previously discussed in 2.1.1. Either all or a part of any other grid plane may be designated to be a boundary. The primary purpose of these boundaries is to subdivide the problem into regions which have the same thermal parameters. These regions are called blocks. Blocks and boundaries are illustrated in Fig. 2. A block is described by giving the values of its high and low index boundaries in the three coordinate directions and defining its thermal parameters.

Blocks are either internal or external. A block is internal if none of its boundaries are coincident. It is external if its two boundaries in the same coordinate direction (i.e., high X, low X, or high Y, low Y, etc.) are coincident. Internal blocks may contain either materials or coolants. External blocks may contain only coolants. They are used to specify the boundary conditions of the problem in terms of the thermal parameters given for the coolants which they contain. Internal blocks contain only internal points. External blocks contain only external points.

2.1.4 Coolant Blocks

Coolant block temperatures are determined by a 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 must flow normal to one of the three coordinate planes as illustrated in Fig. 3. The flow direction may be either positive (in the direction of increasing coordinate value) or negative.

ARROWS INDICATE FLOW
DIRECTION IN COOLANT BLOCKS

Fig. 3. Coolant blocks in TAC3D

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 those 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 block boundaries and by radiation between opposite block boundaries. The radiation is calculated one-dimensionally on the basis of the equations

$$\frac{q_r}{A} = \mathfrak{F}_{1-2} \sigma (T_1^4 - T_2^4) ,$$

$$\mathfrak{F}_{1-2} = \frac{1}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} ,$$

where

q_r is heat transferred by radiation from Surface 1 to Surface 2, Btu/hr
 A is area of the heat transfer surface located at the lowest value of the coordinate perpendicular to the direction of radiant heat flow, ft²

T_1 is temperature of Surface 1, °R

T_2 is temperature of Surface 2, °R

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

$\mathfrak{F}_{1-2}$ is overall radiant interchange factor between Surfaces 1 and 2, dimensionless

ϵ_1 is emissivity of Surface 1, dimensionless

ϵ_2 is emissivity of Surface 2, dimensionless

2.1.5 Gaps

On boundaries between adjacent material blocks, narrow open spaces or gaps may be defined as illustrated in Fig. 2. These gaps contain the gases previously discussed. A gap is specified on the high boundary of a material block in either coordinate direction by giving its thickness and

the number of the gas which it contains. It is implied that a gap is narrow because its thickness should be less than the distance between the boundary and the points adjacent to that boundary.

Heat transfer across a gap is by radiation between its surfaces and by conduction through the gas. This heat transfer is purely one-dimensional. Radiation is calculated on the basis of the same equations as given in Section 2.1.4 for coolant blocks. A gap contains no points and its lower surface is neither a grid plane nor a boundary. A gap is used only to define the thermal resistance which is added in series to that associated with conduction between adjacent points. In performing the calculation for points adjacent to a gap it is assumed that over its thickness the gap has actually replaced the block material. Heat generation, thermal resistance, and capacitance in the block material are adjusted accordingly.

2.1.6 Gap Planes

A gap plane is defined as a boundary any portion of which bounds a coolant block, or along any portion of which there is a gap. From this definition, it follows that the external boundaries are always gap planes.

2.2 INITIAL TEMPERATURES

Since a transient calculation is always performed, the temperatures of the points must be given at the initial time of the calculation.

2.2.1 Specified Initial Temperatures

Three-dimensional space regions enclosing points having the same initial temperatures may easily be defined. This is done by giving the values of the bounding grid planes and the initial temperature. The regions are of two types, depending upon whether they contain internal or external points.

An internal point region is always located by six different grid plane values. There may be as many such regions as there are internal points or there may be as few as one. Every internal point must be part of some region. If regions are overlapped, the last temperatures described for that space apply.

An external point region is located by six grid planes, two of which are coincident and are an external boundary. There may be as many external point regions as there are external points in the problem. All external points need not be included in a region, but those which are not are automatically assigned an initial temperature of 460°R.

2.2.2 Previously Punched Temperatures

If the calculation for a problem is run up to a certain point in time, results at this time may be punched on cards as part of the output. These results include all point temperatures, coolant terminal temperatures and the time. This deck may be used as initial temperature input for a succeeding problem. When so used, it will supersede the specified values described in Section 2.2.1. The basic purpose of the punched output is to enable a long calculation to be performed in several computer runs. However, the punched output of one problem may be used in another problem so long as both have the same number of points in each coordinate direction.

2.3 THE TIME VARIABLE

2.3.1 Initial Time Value

The time scale begins at zero if the initial temperatures are specified by regions as described in Section 2.2.1. If a deck of previously punched temperatures is being used, the time scale begins at the time indicated in hours on the first card of the deck. This time may be changed to zero or any other desired value by re-punching the card.

2.3.2 Time Incrementation

The time scale of the problem is broken down into finite increments called time steps. The array of point temperatures is calculated at times corresponding to the ends of each of these time steps. The calculation for each time step is called an iteration. Groups of equal and adjacent time steps are called time periods. The time incrementation of the problem is specified by giving the end of each time period and the value of the time step which is to be used within it. The beginning of the first time period is the initial time value discussed in Section 2.3.1.

In performing the calculation, the current time (i.e., the time corresponding to the end of the current iteration) is determined as

$$\text{current time} = \text{initial time value} + \sum \text{all preceding time steps} + \text{current time step} .$$

Within any time period, the time step given for that period is used until the current time equals or exceeds the specified end of the period. Therefore, if the length of a time period is not exactly divisible by its time step, the remainder is replaced by one whole time step and the end of the current period is accordingly moved forward from its specified value.

2.3.3 Initial Iteration

Some values within the program are initialized on the first iteration of a problem. This first iteration is always performed before the input time steps are recognized and is over an extremely small time step of 10^{-10} sec. It is not included in the iteration count. It is included in determining the current location on the time scale, but its effect should be negligible.

2.4 THE STEADY-STATE OPTION

Steady-state results are found by performing a transient calculation for a sufficient number of iterations to attain thermal equilibrium. This may be accomplished entirely under the user's control by solving a pseudo-transient problem wherein arbitrary values of the time steps and the volumetric specific heats of the materials are given as input. However, the user does not generally have sufficient information to select values which will produce the steady-state solution efficiently. If, for any given set of specific heats, the time steps are chosen too large, then the unstable condition described in Section 5.1 develops. The problem is terminated and another attempt at its solution must be made using smaller time steps. If the time steps are chosen too small, then an excessively large number of iterations may be required to reach equilibrium.

The inefficiencies described above can usually be avoided by using the steady-state option. Time steps and material specific heat functions need not be given as input. Instead, values designed to affect rapid

convergence are calculated within the code. The following steps are carried out for each steady-state iteration:

1. Determine a specific heat value for each material point. These values are chosen such that all material points will tend to have approximately the same stability characteristics in the solution for material point temperatures.
2. Assign the time step according to an increasing function of time step versus number of iterations performed. Assign a value of one hour for the initial time step.
3. Solve for all temperatures.
4. Inspect the temperature results for instability as evidenced by negative or extremely large values. If instability is found, reduce the time step by dividing it by a factor of 1.5. Allow no further increase in Step 2 above.

Residuals are checked at regularly spaced intervals of completed iterations. The value of the residual, R , for a material point is

$$R = \frac{q_c}{q_{k1} + q_{k2} + q_{k3} + q_{k4} + q_{k5} + q_{k6} + q_g} ,$$

where all terms on the right hand side are absolute values at the point and

q_c is rate of heat storage, Btu/hr

q_g is rate of heat generation, Btu/hr

q_{k1} through q_{k6} are rates of heat conduction from the point to each of its six neighboring points, Btu/hr

The steady-state iterations are terminated when the residual at every material point is less than 10^{-4} . This implies that the transient term is of the order of 10^{-4} smaller than the other terms in the energy equation; thus, the steady-state form of the energy equation essentially applies. Points contained in certain classes of dummy materials are not considered either in setting specific heats or in checking residuals for convergence. Such materials require special identification as described in Section 4.1.1.

Upon completion of the steady-state iterations, a series of twenty-one smoothing iterations is performed. Specific heats remain fixed at their last computed values 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 attained.

Temperature results are printed for the last steady-state iteration and for every seventh smoothing iteration. It is important to inspect these results for any evidence of instability. Residuals are printed for the last smoothing iteration. These should always be inspected to insure that no values have increased to greater than 10^{-4} during the smoothing iterations.

2.5 THERMAL PARAMETER FUNCTIONS

The actual values of the thermal parameters discussed in Section 2.1.2 are given by FORTRAN V arithmetic statement functions. These functions are inserted into the program by recompiling two subroutines, MADATA and FLODAT, each time a problem is run. The functions for materials and gases are inserted into MADATA, and those for coolants are inserted into FLODAT. On the Univac 1108 computer system, it has been found that the cost of recompiling these subroutines is generally not a controlling factor in establishing the total cost of a TAC3D computer run.

2.5.1 Function Names

The function names contain the basic name of the thermal parameter being defined, the member number within the material, coolant or gas set, and a dummy argument X. The names and definitions of the functions which may be specified for each of the three sets are given below. The dash indicates any of the integers 1 through 15 representing the member numbers. Members of the same set must be assigned different numbers whenever any of their corresponding thermal parameters cannot be defined identically. For coolants, the signed flow direction number is a thermal parameter within the scope of this restriction.

Function Name	Definition	Units
<u>MATERIALS</u>		
RCON - (X)	Thermal conductivity in the X (or radial) direction	Btu/hr-ft-°R
ACON - (X)	Thermal conductivity in the Y (or axial) direction	Btu/hr-ft-°R
TCON - (X)	Thermal conductivity in the Z (or theta) direction	Btu/hr-ft-°R
SPEC - (X)	Volumetric specific heat	Btu/ft ³ -°R
EMRH - (X)	Emissivity in the +X (or +radial) direction	dimensionless
EMRL - (X)	Emissivity in the -X (or -radial) direction	dimensionless
EMAH - (X)	Emissivity in the +Y (or +axial) direction	dimensionless
EMAL - (X)	Emissivity in the -Y (or -axial) direction	dimensionless
EMTH - (X)	Emissivity in the +Z (or +theta) direction	dimensionless
EMTL - (X)	Emissivity in the -Z (or -theta) direction	dimensionless
HEAT - (X)	Volumetric heat generation rate	Btu/hr-ft ³
<u>COOLANTS</u>		
SPH - (X)	Specific heat capacity	Btu/lb-°R
REYN - (X)	Reynolds number	dimensionless
H - A(X), H - B(X), H - C(X)	Heat transfer coefficient in a specified low (A), middle (B), or high (C) range of Reynolds number	Btu/hr-ft ² -°R
TIN - A(X), TIN - B(X), TIN - C(X)	Inlet temperature in a specified low (A), middle (B), or high (C) range of one of the variables time, mass flow rate or outlet temperature	°R

Function Name	Definition	Units
<u>COOLANTS (Cont.)</u>		
FLO - A(X), FLO - B(X), FLO - C(X)	Mass flow rate in a specified low (A), middle (B), or high (C) range of one of the variables time, outlet temperature, or inlet temperature	lb/hr
<u>GASES</u>		
GCON - (X)	Thermal conductivity	Btu/hr-ft-°R

2.5.2 Function Variables

The thermal parameter functions may be constants or they may be dependent upon current local values of some of the calculation variables. Definitions of these variables and their allowed uses are given below for each of the three sets.

MATERIALS

Variable Name*	Definition	Units
DR	1. Temperature of a point, or 2. Local temperature of a gap surface, or 3. Local temperature of an internal coolant block boundary	°R
HR	Time	hr
FTR	X (or radial) coordinate of a point	ft
FTZ	Y (or axial) coordinate of a point	ft
FTT	Z (or theta) coordinate of a point	ft (radians for cylindrical coordinates)

*For materials, DR has three different definitions, depending upon the function in which it is to be used.

MATERIALS (Cont.)

Function Name	Allowed Variables*
RCON - (X), ACON - (X), TCON - (X)	DR ¹⁾ , FTR, FTZ, FTT
SPEC - (X)	DR ¹⁾
EMRH - (X) EMRL - (X)	DR ^{2),3)} , FTZ, FTT
EMAH - (X) EMAL - (X)	DR ^{2),3)} , FTR, FTT
EMTH - (X) EMTL - (X)	DR ^{2),3)} , FTR, FTZ
HEAT - (X)	DR ¹⁾ , FTR, FTZ, FTT, HR

COOLANTS

Variable Name	Definition	Units
DR	Temperature of a point in a coolant	°R
HR	time	hr
FTR	X (or radial) coordinate of a point. Only defined for coolants flowing parallel to the X (or radial) axis	ft
FTZ	Y (or axial) coordinate of a point. Only defined for coolants flowing parallel to the Y (or axial) axis	ft
FTT	Z (or theta) coordinate of a point. Only defined for coolants flowing parallel to the Z (or theta) axis	ft (radians for cylindrical coordinates)
ST	Local temperature of a coolant block boundary	°R

*DR is labeled to indicate which of its three above definitions apply for each function.

COOLANTS (Cont.)

Variable Name	Definition	Units
RE	Reynolds number at a point in a coolant	dimensionless
TIN	Inlet temperature of a coolant	°R
TOUT	Outlet temperature of a coolant	°R
FR	Flow rate of a coolant	lb/hr

Function Name	Allowed Variables
SPH - (X)	DR, HR, FTR, FTZ, FTT, TI TIN, TOUT, FR
REYN - (X)	DR, HR, FTR, FTZ, FTT, ST, RE, TIN, TOUT, FR
H - A(X), H - B(X), H - C(X)	DR, HR, FTR, FTZ, FTT, ST, RE, TIN, TOUT, FR
TIN - A(X), TIN - B(X), TIN - C(X)	HR, TOUT, FR
FLO - A(X), FLO - B(X), FLO - C(X)	HR, TIN, TOUT

GASES

Variable Name	Definition	Units
DR	Local linear average of the two surface temperatures of a gap	°R
FTR	X (or radial) coordinate of a point	ft
FTZ	Y (or axial) coordinate of a point	ft
FTT	Z (or theta) coordinate of a point	ft (radians for theta coordinates)

GASES (Cont.)

Function Name	Allowed Variables
GCON - (X)	DR, FTR, FTZ, FTT

2.5.3 Standard Function Values

Not all of the thermal parameters are used in every problem. Functions need not be specified for those which are not used. If no function is specified for a parameter which is used, then that parameter is automatically set equal to a standard value as indicated below.

$$RCON - (X), ACON - (X), TCON - (X) = 10^{-6} \text{ Btu/hr-ft-}^{\circ}\text{R}$$

$$SPEC - (X) = 1. \text{ Btu/ft}^3\text{-}^{\circ}\text{R}$$

$$EMRH - (X), EMRL - (X), EMAH - (X),$$

$$EMAL - (X), EMTH - (X), EMTL - (X) = 10^{-6}, \text{ dimensionless}$$

$$HEAT - (X) = 0.$$

$$SPH - (X) = 1. \text{ Btu/lb-}^{\circ}\text{R}$$

$$REYN - (X) = 0.$$

$$H - A(X), H - B(X), H - C(X) = 10^{-6} \text{ Btu/hr-ft}^2\text{-}^{\circ}\text{R}$$

$$TIN - A(X) = 460. ^{\circ}\text{R}$$

$$TIN - B(X), TIN - C(X) = 0.$$

$$FLO - A(X) = 10^6 \text{ lb/hr}$$

$$FLO - B(X), FLO - C(X) = 10^{-6}$$

$$GCON - (X) = 0.$$

2.5.4 Interconnection of Coolants

Special functions may be used to define inlet temperatures such that the inlet temperature of one coolant is equal to the outlet temperature of another. As an example, the inlet temperatures of Coolants 3, 5, and 2 can be defined as follows:

$$TIN3A(X) = 560.$$

$$TIN5A(X) = T0(3)$$

$$TIN2A(X) = T0(5)$$

The result is that, in the A, or low range of inlet temperature, the inlet of Coolant 3 is 560.°R, the inlet of 5 is the outlet of 3 and the inlet of 2 is the outlet of 5. As few as two and as many as all fifteen coolants may be interconnected in the manner illustrated. Although the example is for the A range of inlet temperature, such interconnections may also be made for the B and C range functions. The blocks containing the interconnected coolants need not necessarily be adjacent. Extreme caution should be used when coolants are interconnected for a transient calculation. A large truncation error can be developed if the time steps are not taken sufficiently small. The error is discussed in Section 5.2.2.2.

2.5.5 Function Control Constants

The need to recompile subroutines MADATA and FLODAT for each different set of thermal parameters imposes a limitation on using the computer to do several different problems, or consecutive cases, in one run. This limitation is partially overcome by the availability of the function control constants A1 through A18, which may be used in the parameter functions. A typical use of these constants is illustrated by the example given below.

Subroutine MADATA:

$$RCON3(X) = A5*(Function\ 1) + A9*(Function\ 2)$$

Problem inputs:

<u>Problem No.</u>	<u>A5</u>	<u>A9</u>
1	1.	0.
2	0.	1.

Effectively, $RCON3(X) = \text{Function 1}$ for Problem 1 and $RCON3(X) = \text{Function 2}$ for Problem 2.

2.5.6 Additional Arithmetic Statement Functions

Additional functions may be defined and used in one or more of the thermal parameter functions. They are arithmetic statement functions containing the dummy argument X. Their use is illustrated in Appendix B. If an additional function is used in material and gas functions and also in

coolant functions, it must be inserted into both Subroutines MADATA and FLODAT.

2.5.7 Function Subprograms

Some desired thermal parameter formulations cannot easily be expressed by the exclusive use of arithmetic statement functions. Step functions are an example. In such cases, function subprograms which contain any legitimate FORTRAN operations may be used, as illustrated in Appendix C.

3. OUTPUT AND OUTPUT CONTROL

3.1 GENERAL

The initial pages of a problem output contain a description of the problem, mostly in tabular form. Also, the input cards are printed, one card per line, to insure the exact reproducibility of the input at a later date. Most of the calculation results are presented as matrices of point conditions lying in planes normal to one of the coordinate directions. Values for individual points are located by index numbers printed along the edges of the matrices. The coordinate value corresponding to each index is given in the problem description. The extreme high and low index values in either coordinate direction represent the external points.

3.2 BASIC OUTPUT

Since the spatial temperature distribution is three-dimensional, the basic output consists of sets of two-dimensional matrices where each matrix in a set gives the point temperatures in a plane normal to one of the coordinate directions. The set of these two-dimensional matrices is sequenced such that the first one printed corresponds to the lowest coordinate value and the last one corresponds to the highest coordinate value. The plane of the two-dimensional matrix is specified by the user.

Printed with each set of matrices are the terminal temperatures and flow rates of the coolants. If the time steps of a problem are given as input, then the frequency of iterations for which basic output will be printed is also given as input. If the problem is being run under the steady-state option, the basic output is printed only for the smoothing iterations described in Section 2.4, and then only for every seventh iteration. Basic output is always printed for the final iteration of a problem regardless of the method under which the calculation is being performed.

All basic output values are converted to integers before they are printed. Temperatures are converted from their calculated real values, RTR, in °R to integers, ITF, in °F by means of the following expression:

$$ITF = RTR - 459.5.$$

The result is that remainders greater than or equal to 0.5°R are rounded upward, while those less than 0.5°R are rounded downward.

In materials, a point temperature is the calculated temperature of the material at the location given by the coordinates of the point. In coolants, a point temperature is a mean effective coolant temperature evaluated over the interval between the two grid planes which bound the point and are normal to the flow direction. The mean effective value is that calculated for the condition of constant wall (or adjacent material block boundary) temperatures over the interval. Therefore, the location of a coolant point temperature along the flow direction is not exactly that given by the point coordinate. However, unless the grid planes are very widely spaced along the flow direction, locating the coolant temperatures by the point coordinates is sufficiently accurate for most engineering purposes.

3.3 AUXILIARY OUTPUT

3.3.1 Overlay

To facilitate interpretation of basic output, a series of overlay forms are printed corresponding to the planes where the output temperature matrices are printed. The boundaries are outlined by a series of the following symbols:

- boundary with gap adjacent
- * boundary with coolant adjacent
- boundary with no gap or coolant adjacent

A scale is used which causes the lines generated to fall between the appropriate point temperatures in each basic output matrix. This diagram can be traced onto a sheet of transparent paper which can then be laid over any basic output matrix. The resulting display contains no dimensional information but is useful for a quick qualitative evaluation of results.

3.3.2 Overall Heat Balance

An overall problem heat balance is always printed for the last iteration of a problem. The volumetric heat generation rate for each material block and the heat gained or lost by each coolant block is itemized. The two quantities are totaled separately. The totals should approach equality as thermal equilibrium is attained. There are exceptions, particularly in problems where dummy materials having extremely high specific heats are used. However, if the two totals differ appreciably, a steady-state solution should not be assumed unless some acceptable explanation can be found.

3.3.3 Residuals

When the steady-state option is being used, residuals for all points in the problem are always printed. They are printed as sets of matrices in the manner described for the point temperatures in Section 3.2. If convergence is obtained, the residuals are those corresponding to the last smoothing iteration. If convergence is not obtained, the last available values are printed.

Residuals for coolant points and certain dummy material points are not considered in determining convergence. Zeros are printed for the residuals at these points.

3.4 OPTIONAL OUTPUT

There are six types of optional output which are described below.

3.4.1 Decimal Temperatures

All basic output is printed in floating point numbers rather than integers. The calculated temperatures, RTR, in °R are converted to the printed values, RTF, in °F by means of the following expression:

$$RTF = RTR - 460.0 \quad .$$

Even when the decimal temperature option is not used, basic output in floating point numbers as well as in integers is printed for the last iteration of the problem.

3.4.2 Geometry Factors, Effective Conductivities and Effective Conductances

The heat rate between Points 1 and 2 in each coordinate direction is given by

$$q = K(T_1 - T_2)$$

where

q is heat rate, Btu/hr

T is point temperature, °F

K is thermal conductance, Btu/hr-°F.

K may be factored

$$K = (G)(C)$$

where

G is component of K which depends purely on geometry, ft

C is component of K which depends on temperatures and thermal parameters, Btu/hr-ft-°F.

G is called the geometry factor. C is called the effective conductivity.

Values of G , C , and K are printed as sets of matrices in the manner described for the point temperatures in Section 3.2. Each matrix corresponds to the points in a plane normal to one of the coordinate directions. The value given for a point is the value between that point and its next higher indexed neighbor. C and K may be printed either for the last iteration only or for all iterations where basic output is printed.

If the option of printing G is used, nodal volumes of all the points are also printed as a set of matrices.

3.4.3 Heat Rates and Heat Fluxes

The heat rates defined in Section 3.4.2 and the associated heat fluxes are printed in the same form as the effective conductivities. The heat fluxes, q'' , are determined from the heat rates by

$$q'' = \frac{q}{A}$$

where A = heat flow area at the grid plane between points, ft^2 .

Heat rates and heat fluxes may be printed either for the last iteration only or for all iterations where basic output is printed.

3.4.4 Surface Temperatures

Surface temperatures are printed as integers using the conversion previously given for the point temperatures. The values printed are the temperatures of the gap planes* which are external boundaries and the temperatures of the portions of gap planes which are adjacent to gaps or coolant blocks. For the gap planes which are external boundaries, a full matrix of surface temperatures is printed corresponding to all the points in the gap plane. For gap planes adjacent to gaps or coolant blocks, surface temperatures are printed only for those points that lie in the portion of the gap plane comprising the boundary of the gap or coolant block.

In the case of a gap plane adjacent to a gap, there are two surface temperatures at any location along the gap plane. They are printed as a pair of numbers in a single matrix location. No surface temperatures are calculated over portions of gap planes which are not actually adjacent to a gap or a coolant. These locations are therefore blank in the printed matrices. Surface temperatures may be printed either for the last iteration only or for all iterations where basic output is printed.

3.4.5 Punched Output

The point temperatures and coolant terminal temperatures of the last iteration may be obtained as a deck of punched cards. This card deck may then be used as the initial temperature input for another computer run, as described in Section 2.2.2.

If convergence is not attained for a problem being run under the steady-state option, the last available set of temperatures is punched automatically. For all other cases, punched temperatures are produced only if the punch option is used.

Card images of the punched temperature deck are printed in the output.

*Gap planes are special cases of grid planes as defined in Section 2.1.6.

3.4.6 Tape Output

A tape containing problem geometry and temperature data may be created as a part of the output. The data and its arrangement on the tape are described in Appendix F. The tape option has been designed primarily to automate the input of code results to plotting routines.

3.5 SPECIAL OUTPUT

The program contains a subroutine named CUSTOM which is always called immediately after the printing of basic output. This subroutine contains all the common elements of the program but has no executable statements other than RETURN. By compiling the required FORTRAN instructions into CUSTOM, calculations using the variables in common may be performed and the results printed. No variables in COMMON should be re-defined in CUSTOM. See Element CDE in the program listing of Ref. 2 for the names of variables in COMMON.

4. INPUT FORMULATION

The input is subdivided into data groups. Instructions for formulating each of these groups are given in Section 4.1. Instructions for arranging the groups into a "ready to run" card input are given in Section 4.2.

4.1 DATA GROUPS

Card descriptions, pertinent dimensional limits, and special notes applicable to setting up each of the input groups are given in terms of the symbols IQ, JQ, KQ, IGQ, JGQ, KGQ, and LQ. The definitions of these symbols and their assigned values are discussed in Appendix D.

4.1.1 Thermal Parameter Functions (See all of Section 2.5.)

Card Description

The arithmetic statement functions are given on individual cards (and continuations if required) according to the rules of FORTRAN V. This data group may contain only the arithmetic statement functions described in Section 2.5, and comment cards. The group of functions for materials and gases may be arranged in any order. The group of functions for coolants may also be arranged in any order. An exception is that in either group a statement function used on the right hand side of an equal sign must have been defined previously within that group.

Limits on the Number of Functions

No arithmetic statement functions need be given. In this case, all thermal parameters relevant to the problem will have the constant standard values defined in Section 2.5.3. However, the FOR cards for MADATA and FLODAT must be included (see Section 4.2) so that these subroutines will be compiled.

There may be more arithmetic statement functions than those for the 345 function names defined in Section 2.5.1. This can occur when additional arithmetic statement functions are used as described in Section 2.5.6.

Therefore, the allowable maximum number of arithmetic statement functions cannot be specified.

NOTES:

1. Only those thermal parameters which are relevant to the problem need be defined.
2. Radiation between a set of internal coolant block boundaries is totally excluded from the calculation whenever the emissivity of the material adjacent to either boundary is 10^{-6} (the standard function value) or less. By "totally excluded" it is meant that the calculation is performed using an algorithm which takes no account of radiation.
3. If the steady-state option is being used, the material volumetric specific heat functions, SPEC - (X), may be omitted. If included, they have no effect on the problem results. There are exceptions to the above rule for two classes of dummy materials:
 - a. Materials which are intended to retain their initial temperatures by virtue of a very high heat capacity
 - b. Materials which are intended to act as nearly perfect insulators by virtue of thermal conductivities in both coordinate directions which approach zero

For both of the above, the function SPEC - (X) must be assigned a value of 10^6 Btu/ft³-°R or greater.

4. If the steady-state option is being used, no thermal parameter function may be time dependent.

4.1.2 Titles

Each title card represents one line of title information. The title cards are printed at the beginning of the problem output in the order of

their input sequence. The first title card is printed at the top of every output page.

Card Description

Title cards may contain any alphanumeric information in columns 1-72.

Limits

	<u>Minimum</u>	<u>Maximum</u>
Number of title cards	1	No limit

NOTE:

Each title card must contain at least one character in Columns 1-72.

4.1.3 Indicators

Certain items of information required to define the problem and specify the desired output are given by means of the indicator words and word groups described below.

Coordinate System Indicator

One of the two coordinate systems shown in Fig. 1 must be specified by means of the following indicator word groups.

<u>Indicator</u>	<u>Coordinate System</u>
RECTANGULAR GEOMETRY	Cartesian
CYLINDRICAL GEOMETRY	Cylindrical

Print Indicator

The presentation of three-dimensional results is discussed in Chapter 3. They are printed as sets of two-dimensional matrices. The code user specifies the planes of these matrices by means of one of the following indicator word groups.

<u>Indicator</u>	<u>Printing Scheme</u>
RADIAL PRINT	Print planes of constant X (or radial) coordinate

<u>Indicator</u>	<u>Printing Scheme</u>
AXIAL PRINT	Print planes of constant Y (or axial) coordinate
THETA PRINT	Print planes of constant Z (or theta) coordinate

If no indicator is given, planes of constant Z (or theta) coordinate will be printed.

Option Indicators

The steady-state option described in Section 2.4 and any of the output options described in Sections 3.4 and 3.5 may be specified by means of the following indicators.

<u>Indicator</u>	<u>Option</u>
STEADY-STATE	Use the steady-state option; calculate the steady-state solution only.
ALL DECIMAL TEMPERATURES	Print decimal rather than integer temperatures for all iterations where basic output is printed.
RESISTANCES	Print geometric factors and nodal volumes.
CONDUCTIVITIES	Print effective conductivities and effective conductances for the last iteration.
ALL CONDUCTIVITIES	Print effective conductivities and effective conductances for all iterations where basic output is printed.
HEAT FLUXES	Print heat rates and heat fluxes for the last iteration.
ALL HEAT FLUXES	Print heat rates and heat fluxes for all iterations where basic output is printed.
SURFACE TEMPERATURES	Print surface temperatures for the last iteration.
ALL SURFACE TEMPERATURES	Print surface temperatures for all iterations where basic output is printed.

<u>Indicator</u>	<u>Option</u>
PUNCH	Punch temperatures for the last iteration.
TAPE	Prepare a tape containing problem geometric data and results.

Card Description

There is one indicator per card, starting in Column 7 and spelled exactly as given above. The cards may be arranged in any order.

Limits

The coordinate system indicator must always be given. All or none of the other indicators may be given.

4.1.4 X Grid Planes (See Section 2.1.1.)

An X (radial) grid plane is normal to the X (radial) coordinate axis. Each X (radial) grid plane is defined by giving its X (radial) coordinate value in inches.

Card Description

Give the X grid planes in ascending sequence. Use format 6E12.4. Use as many cards as required.

Limits

	<u>Minimum</u>	<u>Maximum</u>
Number of X grid planes	3	(IQ-1)

4.1.5 Y Grid Planes (See Section 2.1.1.)

A Y (axial) grid plane is normal to the Y (axial) coordinate axis. Each Y (axial) grid plane is defined by giving its coordinate value in inches.

Card Description

Give the Y grid planes in ascending sequence. Use Format 6E12.4. Use as many cards as required.

Limits

	<u>Minimum</u>	<u>Maximum</u>
Number of Y grid planes	3	(JQ-1)

4.1.6 Z Grid Planes (See Section 2.1.1.)

A Z (theta) grid plane is normal to the Z (theta) coordinate. Each Z (theta) grid plane is defined by giving its Z (theta) coordinate value in inches (or degrees).

Card Description

Give the Z grid planes in ascending sequence. Use Format 6E12.4. Use as many cards as required.

Limits

	<u>Minimum</u>	<u>Maximum</u>
Number of Z grid planes	3	(KQ-1)

NOTE:

Theta grid planes may lie in the range $0^\circ \leq \theta \leq 360^\circ$: however, there is no connection between the points in the grid plane $\theta = 0^\circ$ and the corresponding points in the grid plane $\theta = 360^\circ$.

4.1.7 Block Information (See Sections 2.1.3 through 2.1.6 and Section 2.5.1.)

A block is described by the following data:

1. Material number, any of 1.0 through 15.0, for material blocks
2. Negative coolant number, any of -1.0 through -15.0 for coolant blocks
3. Flow direction number for coolant blocks
4. The grid plane values of its six boundaries given in inches or in degrees for the theta coordinate

5. Gaps located on the high block boundaries in each of the coordinate directions

- a. Gap thickness in inches, or in degrees for the theta coordinate
- b. Gas number, any of 1.0 through 15.0, for the gas contained in the gap

The flow direction number, Item 3 above, is one of the following indicators:

<u>Indicator</u>	<u>Flow Direction</u>
1.0	Parallel to the X (radial) axis
2.0	Parallel to the Y (axial) axis
3.0	Parallel to the Z (theta) axis

The indicators are positive when the coolant is flowing in the positive coordinate direction. They are negative for flows of opposite direction.

Blocks must be specified such that every internal point is contained in some internal block and every external point is contained in some external block.

Card Description

The data of each block are given on a set of three cards. Format 6E12.4 is used. The sets may be arranged in any order.

First Card

Columns

1-12	Material number or negative coolant number
13-24	Flow direction number for a coolant block; not used for a material block
25-72	Not used

Second Card

Columns

1-12	Low X (radial) boundary
13-24	High X (radial) boundary
25-36	Low Y (axial) boundary

37-48	High Y (axial) boundary
49-60	Low Z (theta) boundary
61-72	High Z (theta) boundary

For external blocks, the high and low boundaries of one coordinate direction are the same and are an external boundary.

Third Card

Columns

1-12	X (radial) gap thickness on the high X boundary
13-24	X (radial) gap gas number, any of 1 through 15
25-36	Y (axial) gap thickness on the high Y boundary
37-48	Y (axial) gap gas number, any of 1 through 15
49-60	Z (theta) gap thickness on the high Z boundary
61-72	Z (theta) gap gas number, any of 1 through 15

For any high block boundary where there is no gap, the words for the gap thickness and gas number are left blank.

Limits

	<u>Minimum</u>	<u>Maximum</u>
Number of blocks:		
Total	7	LQ
Internal	1	LQ less number of external blocks
External	6	LQ less number of internal blocks
Number of block boundaries in the X (radial) direction	2	(IQ-1)

Number of block boundaries in the Y (axial) direction	2	(JQ-1)
Number of block boundaries in the Z (theta) direction	2	(KQ-1)
Number of gap planes in the X (radial) direction	2	IGQ
Number of gap planes in the Y (axial) direction	2	JGQ
Number of gap planes in the Z (theta) direction	2	KGQ
Number of materials	1	15
Number of coolants	3	15
Number of gases	0	15

NOTES:*

1. External blocks must be specified so as to completely cover all six external boundaries. The external blocks must contain only coolants. This includes the cases of the low radial boundary at 0 inches and of the high and low theta boundaries on the coincidence of 0° and 360° .
2. Materials, gases, and coolants must be assigned different numbers whenever any of their corresponding thermal parameters cannot be defined identically. For coolants, the signed flow direction number is a thermal parameter within the scope of this limitation.
3. The thickness of a gap must be less than the distance between the high block boundary at which it is defined and the grid plane adjacent to the low side of that boundary. This limitation is illustrated in

*The information in some notes was previously given in Section 2 and is restated here to help prevent errors in input formulation.

Item a of Fig. 4. Preferably, t should be much less than $d/2$.

4. There may be no gaps on an external boundary or on a boundary adjacent to an internal coolant.
5. Two internal coolant blocks having the same flow direction may not have a common boundary which is parallel to that flow direction, as illustrated in Item b of Fig. 4. Within the scope of this limitation, two flow directions are considered the same even if they differ only in sign.
6. A coolant block must contain at least one grid plane perpendicular to the flow direction, as illustrated in Item c of Fig. 4.
7. For internal coolant blocks, heat transfer occurs only at the block boundaries which are parallel to the flow direction. The two boundaries perpendicular to the flow direction are adiabatic, as illustrated in Item d of Fig. 4.
8. The coolant in an internal coolant block may have a heat transfer coefficient, $H_{--}(X)$, which is defined as a function of surface temperature ST . If two opposite boundaries of the block are separated by at least two sets of points, the temperatures of these two boundaries will be used to calculate separate heat transfer coefficients for each boundary. If two opposite boundaries of the block are separated by only one set of points, the average of the two boundary temperatures will be used to calculate one heat transfer coefficient which is applied to both boundaries. This limitation is illustrated in Item e of Fig. 4.
9. The same numbered coolant may not be used in two different blocks unless its specific heat is a constant value.

(a) LIMITATION ON GAP THICKNESS

(b) LIMITATION ON ADJACENT INTERNAL COOLANTS

(c) MINIMUM GRID PLANE REQUIREMENT FOR COOLANT BLOCKS

(d) ADIABATIC BOUNDARIES OF AN INTERNAL COOLANT BLOCK

(e) LIMITATION ON USING HEAT TRANSFER COEFFICIENT AS A FUNCTION OF SURFACE TEMPERATURE FOR INTERNAL COOLANT BLOCKS
 $H--(X)$ IS A FUNCTION OF ST

Fig. 4. Gap and coolant limitations illustrated in a cross-section normal to the Z axis

10. If the same numbered coolant is used in different blocks, each block will have its own inlet temperature as determined by the inlet temperature functions given for the coolant. This holds true even when the blocks are adjacent.
11. Coolant flow rates and terminal temperatures printed in the output are identified by coolant number only. If the same numbered coolant is used in two different blocks, then the results printed are those for that coolant in the block which was given last in the block description input card sequence.

4.1.8 Specified Initial Temperatures (See Section 2.2.1.)

Each initial temperature space is specified by giving the values of its bounding grid planes in inches (or in degrees for the theta coordinate) and its initial temperature in °F. All internal points must be contained within an internal point space. Only external points for which an initial temperature other than 460°R is desired need be contained within an external point space.

Card Description

The data of each initial temperature space are given on a set of two cards. Format 6E12.4 is used. The sets may be arranged in any order.

First Card

Columns

1-12	Low X (radial) bounding grid plane
13-24	High X (radial) bounding grid plane
25-36	Low Y (axial) bounding grid plane
37-48	High Y (axial) bounding grid plane
49-60	Low Z (theta) bounding grid plane
61-72	High Z (theta) bounding grid plane

For external point spaces, one set of high and low bounding grid planes are the same and are an external boundary.

Second Card

Columns

1-12	Initial temperature of the region, °F
13-72	Not used

Limits

	<u>Minimum</u>	<u>Maximum</u>
Number of internal point spaces	1	Number of internal points
Number of external point spaces	0	Number of external points

NOTES:

1. If the previously punched output of another computer run is used as input, specified initial temperatures must still be given. However, the values actually used in performing the calculation will be the previously punched temperatures.
2. If the steady-state option is being used, either previously punched output or specified initial temperatures must be given.

4.1.9 Coolant Limits (See Section 2.5.1 and 2.5.2 under coolants.)

Under the coolant thermal parameter function names given in Section 2.5.1, it is seen that heat transfer coefficient, mass flow rate, and inlet temperature each have three separate names, as, for example, the names H-A(X), H-B(X), and H-C(X) assigned to the heat transfer coefficient. The purpose of these names is to permit the functions to be defined differently in three ranges of an allowed variable. The variables and the limits of their ranges are specified in this data group. An indicator, called a dependence number, is used to identify the variable for which the ranges are being given.

Card Description

The dependence numbers and limits for each coolant are given on a set of three cards. There must be one set for each coolant. Format 6E12.4 is used.

First Card

This card identifies the coolant and gives the limits for functions H-A(X), H-B(X), and H-C(X). Reynolds number is the only allowed variable. Therefore, no dependence number is required.

Columns

1-12	Absolute value of coolant number
13-24	Reynolds number Limit 1
25-36	Reynolds number Limit 2
37-48	Reynolds number Limit 3
49-60	Reynolds number Limit 4
61-72	Not used

The Reynolds number limits must be given in ascending sequence. All four limits must always be given even though all three function names for the heat transfer coefficient are not used. See Note 1.

Second Card

This card gives the limits for the functions FLO-A(X), FLO-B(X) and FLO-C(X), and identifies the variable for which these limits are being given. The variable is identified by assigning one of the following values to the flow dependence number:

<u>Flow Dependence Number</u>	<u>Meaning</u>
0.0	No limits given; only the function FLO-A(X) is to be used
1.0	The limits are values of current time in hours
2.0	The limits are values of coolant outlet temperature in °F
3.0	The limits are values of coolant inlet temperature in °F

The arrangement of values on the second card is:

<u>Columns</u>	
1-12	Flow dependence number
13-24	Limit 1
25-36	Limit 2
37-48	Limit 3
49-60	Limit 4
61-72	Not used

If the flow dependence number is 0.0, Columns 13-72 are left blank.

Third Card

This card gives the limits for the functions TIN-A(X), TIN-B(X), and TIN-C(X) and identifies the variable for which these limits are being given. The variable is identified by assigning one of the following values to the inlet temperature dependence number:

<u>Inlet Temperature Dependence Number</u>	<u>Meaning</u>
0.0	No limits given; only the function TIN-A(X) is to be used
1.0	The limits are values of current time in hours
2.0	The limits are values of coolant flow rate in lb/hr
3.0	The limits are values of coolant outlet temperature in °F.

The arrangement of values on the third card is:

<u>Columns</u>	
1-12	Inlet temperature dependence number
13-24	Limit 1
25-36	Limit 2
37-48	Limit 3
49-60	Limit 4
61-72	Not used

If the inlet temperature dependence number is 0.0, Columns 13-72 are left blank.

A set of three coolant limit cards must be given for each coolant which is used in the problem. The sets may be arranged in any order.

Limits

	<u>Minimum</u>	<u>Maximum</u>
Number of sets of coolant limit cards	2	15

NOTES:

1. Frequently it is desired to use only one range for the heat transfer coefficient of a given coolant. In this case, input can be minimized by utilizing the standard value of the Reynolds number function. On the first card specify 0.0, 1.0, 2.0, and 3.0 for Reynolds number Limits 1, 2, 3, and 4, respectively. In the coolant thermal parameter function input, define $H-A(X)$ as desired. Give no definition for Reynolds number so that the standard value, $REYN-(X) = 0.0$, applies. The Reynolds number will always lie in the low range and the heat transfer coefficient will therefore always be defined by $H-A(X)$.
2. The coolant inlet and exit temperature limits are given in $^{\circ}F$. When formulating the functions $FLO-A, B, C(X)$ and $TIN-A, B, C(X)$, it should be remembered that the variables TIN and $TOUT$, whose ranges are defined by these limits, are in $^{\circ}R$.

4.1.10 Time History (See Sections 2.3 and 3.1.)

The purposes of this data group are to specify the time incrementation of the problem and to designate the iterations where basic output is to be printed. If the steady-state option is used, this data group may be omitted. If included, the data group is ignored in performing the calculation. The time incrementation is given by defining the time periods as discussed in

Section 2.3.2. The time units used are identified by means of the following indicators:

<u>Indicator</u>	<u>Units</u>
1.0	Seconds
2.0	Minutes
3.0	Hours

The iterations where basic output is to be printed are designated by giving a print frequency number for each time period. If the print frequency number for a time period is X.0, then basic output is printed every Xth iteration within that period. In addition, basic output is printed for the last iteration performed within each time period unless the print frequency for that period is 0.0. If the print frequency for all time periods is 0.0, then no basic output is printed except that for the very last iteration of the problem. It is recommended that the length of each time period be made exactly divisible by its time step. If this is not done, the ends of the time periods will be adjusted as described in Section 2.3.2, making it difficult to obtain basic output at desired times.

Card Description

Each time period is described on a single card. Format 6E12.4 is used.

Columns

1-12	End of period
13-24	Time step
25-36	Print frequency number
37-48	Time unit indicator
49-72	Not used

The cards must be arranged in ascending time sequence.

Limits

	<u>Minimum</u>	<u>Maximum</u>
Number of time periods	1	20

NOTES:

1. All of Section 5 should be read before formulating the time history input for a problem.
2. If a problem is to be continued in another computer run using previously punched temperatures, the time period cards of the preceding run may, but need not, be removed.
3. If the steady-state option is used, the time history data group may be omitted. If time period cards are inadvertently included in the input, they are ignored.

4.1.11 Function Control Constants (See Section 2.5.5.)

Card Description

Give the function control constants in the sequence A1 through A18. Use Format 6E12.4. Leave blank columns for any constant which is not used. There must always be three cards. They will be blank cards if no constants are used.

4.1.12 Previously Punched Temperatures (See Sections 2.2.2, 2.3.1, and 3.4.5.)

This data group is included only when it is desired to supersede the specified initial temperature input described in Section 4.1.8 with the point and coolant terminal temperature results of a preceding calculation. Although the required input cards are punched by the computer system, it may be desired to change individual values on these cards. Therefore, a card description is given.

Card Description

The deck punched by the computer system consists of three card sets arranged in the same order in which they are described below.

Card Set 1

There is only one card in the first card set. The format is E12.6, 4F12.1.

Columns

1-12	Time, hours (Notes 1 and 2)
13-24	Number of X (radial) points in the problem
25-36	Number of Y (axial) points in the problem
37-48	Number of Z (theta) points in the problem
49-60	Iteration number (Notes 1 and 2)

Card Set 2

The values contained in the second card set are the temperatures in °R of all the points, both internal and external, in the problem. There are as many cards as required to list all of the temperatures using Format 6E12.6. The point temperature array is called T(I,J,K) where I is the X (radial) point index, J is the Y (axial) point index, and K is the Z (theta) point index. The values in the T(I,J,K) array are listed continuously with the index I varying most frequently and K varying least frequently.

Card Set 3

The values contained in the third card set are the coolant terminal temperatures in °R. Inlet and outlet temperatures are given for all fifteen coolants with values of 0.0°R being given for those which are not included in the problem. Format 6E12.6 is used, and it follows that there are always six cards in this set. The inlet temperatures of Coolants 1 through 15 are listed in order on the first three cards and their outlet temperatures are listed similarly on the last three.

NOTES:

1. The time and the iteration number given in the first card set are the values at the final iteration of the problem from which the previously punched temperatures were obtained. The time given here is the initial time for the problem in which the previously punched temperatures are to be used. The time history data group described in Section 4.1.10 must contain time periods which end at times greater than this value.

Both the time and iteration number may be set to any desired value by repunching the card on which they appear.

2. If a punched temperature deck is part of the input for a problem to be run under the steady-state option, the time and iteration number must both be zero. Zeros will always be punched for these values when the punching is done under the steady-state option.
3. The previously punched temperature deck ends with five blank cards. They are included to separate it from other jobs which may precede it through the punching equipment. Remove these blank cards before using the deck.

4.2 CARD INPUT FOR A COMPUTER RUN

The actual input for a computer run is set up by arranging control cards, blank cards, and the previously described data groups in the sequence given below. The control cards shown are those for running on the Univac 1108 computer from a Program Complex File tape number XXXX which contains on its first file the processed map, the symbolic elements for Subroutines MADATA and FLODAT, and the relocatable elements for all other code routines. The left hand margin represents Column 1 of these control cards.

∇	ASG	A=XXXX	TAC3D PROGRAM TAPE	
∇D	ASG	DRUM2=10000		(Note 1)
∇D	ASG	DRUM3=10000		
∇D	ASG	DRUM4=10000		
∇	XQT	CUR		
	IN	A		(Note 2)
	TRI	A		
∇P	HGD	THERMAL PARAMETER FUNCTIONS		
∇INW	FOR,*	MADATA/S, MADATA/S, MADATA/R		(Note 3)
-10				
Material and gas thermal parameter functions			(4.1.1 and Note 4)	
∇INW	FOR,*	FLODAT/S, FLODAT/S, FLODAT/R		(Note 3)

Coolant thermal parameter functions	(4.1.1)
Function subprograms, if used	(Appendix C)
VP HDG any alphanumeric heading in Cols. 13-72	(Note 5)
∇ XQT TAC3D/AR	(Note 6)
Titles	(4.1.2)
Blank card	
Indicators	(4.1.3)
Blank card	
X (radial) grid planes	(4.1.4)
Blank card	
Y (axial) grid planes	(4.1.5)
Blank card	
Z (theta) grid planes	(4.1.6)
Blank card	
Block information	(4.1.7)
Blank Card	
Specified initial temperatures	(4.1.8)
Blank card	
Coolant limits	(4.1.9)
Blank card	
Time history	(4.1.10 and Note 4)
Blank card	
Function control constants	(4.1.11)
Blank card	
Previously punched temperatures (if used)	(4.1.12 and Note 7)
Blank card	(Note 8)

NOTES:

1. The three drum assign cards are required only when the steady-state option is being used. If they are included when the steady-state option is not being used, there will be an unnecessary computer charge for requested drum usage.
2. If Subroutine CUSTOM is being used (see Section 3.5), then one additional "IN A" card (a total of two)

must be included at this point. The second "IN A" card is required to bring the second file of the program tape, which contains the symbolic element for CUSTOM, into the Program Complex file.

3. The FOR cards for both Subroutines MADATA and FLODAT must always be included, even in a case where no thermal parameter functions are given.
4. If the steady-state option is being used, both the material volumetric specific heat functions SPEC-(X) (except as discussed in Section 4.1.1, Note 3) and the time history data group may be omitted. If included, they will be ignored.
5. This heading card causes the specified heading to be printed along a single line at the top of every output page produced after program execution. Printed on the next line will be the first problem title card.
6. The name TAC3D/AR appearing on the execute card is the name of the processed map in the standard version of the code. For special versions this name may be different. The name on the execute card must be the same as the name of the processed map on the version being used.
7. The location for the previously punched temperature cards contains no cards if previously punched temperatures are not used.
8. If consecutive problems are to be run, the first title card for the second problem follows the last blank card shown above. A complete set of data (i.e., first title card through last blank card) must be given for the second and subsequent problems. The only similarity requirement among consecutive

problems is that the thermal parameter functions be applicable to all. The number of consecutive problems is limited only by the specified computer running time.

4.3 PERFORMANCE OF A COMPUTER RUN

4.3.1 Execution Time and Printed Page Limits for Use with the Univac 1108 Computer

The maximum allowed execution time and the maximum number of pages of printed output must be included in the run request for a problem. A non-zero warning time may, but need not, be given. These values are specified on the computer run request.

The program calls the computer library subroutine WARN before beginning each iteration. In WARN, the elapsed running time is compared with the allowed execution time. When the difference between the two becomes less than the user specified warning time, no further iterations are performed. Instead, the program is directed to preparation of basic and specified optional output (including punched temperatures) for the current iteration. A warning time of 10 seconds should be adequate to allow for preparation of all possible output. Note that the warning time is effectively subtracted from the allowed execution time. If, for instance, warning and allowed execution times of 10 and 300 seconds, respectively, are given, the time available to complete the calculation is only 290 seconds. It is good practice to give a warning time so that none of the available results are lost if the required execution time exceeds the maximum specified.

Through use of the library subroutine PAGLIM, some provision is made for recovering results, the printing of which would fall beyond the specified maximum number of output pages. Whenever the calculation approaches a point where the page limit will be exceeded, PAGLIM is called by the program, thus causing the punched output for the current iteration to be prepared.

4.3.2 Error Messages

If the calculations of a problem cannot be performed according to the logic of the code, error messages are printed. These are of two types: system error messages and program error messages. System error messages on the Univac 1108 give a description of some abnormal condition within the computer system. An example is the words "floating point overflow at" followed by an octal number which is the instruction address where the overflow occurred. Program error messages originate from write statements within the program itself. They are printed when some rule of input formulation is violated and when difficulty is encountered in performing the calculations. Only identifiers consisting of the name of the subroutine from which the message was written and the message number are printed. These identifiers are related to error descriptions in Appendix A.

5. TRUNCATION AND ROUND OFF ERRORS

5.1 STABILITY OF SOLUTION

5.1.1 Basic Considerations

The solution of the matrix of point temperatures becomes unstable when the temperature change at some point is greater than that required for thermal equilibrium with its six neighboring points. The stability criterion for explicit methods of solution is

$$\Delta t_i = \frac{C_i}{\sum_{j=1}^6 K_{ji}}$$

where

Δt_i is the maximum stable time step for Point i, hr

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

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

In TAC, an implicit method of solution is used for which the stability criterion cannot be precisely defined. It can only be stated that the stable time step is proportional to the same parameter as for an explicit formulation. That is,

$$\Delta t_i \propto \frac{C_i}{\sum_{j=1}^6 K_{ji}} .$$

As is discussed in Section 5.2.2.3, the temperature calculation for the points in coolants is decoupled from that for the points in materials. The implicit solution is performed only for the material point temperatures. The coolant point temperatures are calculated after the material point

temperatures and this calculation has no direct influence on stability. Therefore, the concept of a maximum stable time step is directly applicable only to the material points.

If the time step Δt is sufficiently greater than Δt_i , the error due to instability at Point i appears as a diverging oscillation. Although the value of Δt at which this occurs cannot be predicted for the method of solution used in TAC3D, it is referred to here as Δt^* .

A most desirable approach to the stability problem would be to keep time steps less than the minimum value of Δt_i among all material points in the problem. This approach is not possible because Δt_i values cannot be computed beforehand for the implicit method of solution used. However, the previously stated proportionality provides some qualitative basis upon which it may be endeavored to make all Δt_i values as large as possible in designing the calculation model for any given problem.

5.1.2 Practical Approaches

In steady-state calculations, the transient solution is not of interest. Stability related errors which may occur because Δt exceeds some Δt_i are not important so long as a solution which satisfies thermal equilibrium is finally obtained. The only concern is that Δt always remains less than Δt^* so that diverging oscillations do not develop and cause the calculation to be terminated before reaching steady-state. The code includes a steady-state option which may be used to obtain steady-state solutions efficiently. Under this option, described in Section 2.4, values of Δt approaching Δt^* are used to perform the transient calculation up to a point where the approximate steady-state solution has been reached. Successive values of Δt are increased for each iteration according to some prescribed function. Δt increases unless current results indicate that diverging oscillations are beginning to occur. In this case, Δt is reduced.

The stability related error which is incurred when Δt exceeds Δt_i at Points i is a part of the transient solution. Therefore, if the transient solution is of interest, Δt must be chosen sufficiently small that this error becomes negligible. As previously stated, the Δt_i values cannot be

calculated for the implicit method of solution used in TAC3D. However, the initial minimum value of Δt_i over the points i must be less than the initial value of Δt^* , which can be bracketed for any problem by running a time test case. The problem is run, starting at time zero, through successive time periods each of which is subdivided by a larger time step than that of the period which precedes it. A time history suggested by experience is:

<u>End of Time Period</u>	<u>Time Step</u>
0.01 sec	0.001 sec
0.1 sec	0.01 sec
1.0 sec	0.1 sec
10.0 sec	1.0 sec
100.0 sec	10.0 sec
10.0 min	1.0 min
30.0 min	2.0 min
80.0 min	5.0 min

Within one of these time periods diverging oscillations should develop and terminate the problem. A dump of the common element MISCXX is specified as described in Appendix E and from this dump the time period within which the problem terminated can be identified. The time step values of this and the preceding time period are the limits of a range within which the initial value of Δt^* should lie. It is recommended that the problem be run with time steps no greater than one-third the mean value of these two limits. If thermal parameters which determine the C_i and K_{ji} in the proportionality of 5.1.1 are strongly temperature and/or time dependent, it is possible that both Δt^* and the Δt_i can decrease as the calculation progresses. In such cases, the time step value indicated above should be gradually reduced throughout the calculation. The procedure outlined above is intended only to provide an approximate estimate of the maximum time step which may be used. Assurance that errors related to stability are negligible within a transient solution can be obtained only by re-running the problem with smaller time steps until the results obtained at corresponding times do not differ significantly.

5.2 FINITE DIFFERENCE FORMULATION

All results contain errors because spatially dependent quantities are treated as constant over finite intervals rather than as continuously distributed. Transient results contain further errors because time dependent quantities are treated in the same manner. The sources of error which affect calculation results most significantly are discussed below. For a more detailed discussion, see any standard reference, such as Ref. 5. Errors due to all of these sources become negligible if time steps and/or grid spacing are made sufficiently small.

5.2.1 All Results

5.2.1.1 Grid Spacing

There is always an error which is due to the fact that thermal conductances, thermal capacitances, and heat sources are considered as being concentrated at the calculation points. This error cannot be described in general terms.

5.2.2 Transient Results

5.2.2.1 Nonlinearity of Time

The heat balance equations which are solved to obtain the material point temperatures are formulated under the assumption that the point temperatures vary linearly with time over each time step. The error in the transient solution due to this assumption remains negligible only if the time steps are taken sufficiently small.

5.2.2.2 Evaluation of Time and Temperature Dependent Quantities

Time and temperature dependent components of the conductances, capacitances, and heat sources are evaluated only once for each iteration. Time dependent quantities are evaluated at the time midpoint of the iteration. Temperature dependent quantities to be used in a given iteration are evaluated from the temperatures at the beginning of the iteration (i.e., the temperature results of the preceding iteration).

Particular caution should be used when a coolant inlet temperature is a function of a coolant exit temperature since the two will be out of phase by the amount which the exit temperature changes over the current iteration. If N coolants are interconnected as discussed in Section 2.5.4, the inlet temperature of the Nth coolant will be N iterations out of phase with the current iteration. Large errors can develop unless the time steps are taken appropriately small. As explained in Section 2.3.3, some variables have arbitrary initial values on the first iteration of a problem, and this first iteration is automatically taken over an extremely small time step. Among these variables are coolant inlet temperatures and coolant block boundary temperatures. From this and the preceding discussion, it follows that if there are N interconnected coolants, then N iterations must be performed before all have a real inlet temperature. Therefore, the time step input should include N-1 extremely small (i.e., 1×10^{-10} sec time step) iterations. As explained in Section 2.2.2, a previously punched temperature distribution includes real initial values for coolant inlet and exit temperatures. When previously punched temperature input is included for a problem with interconnected coolants, these additional small initial iterations are not required.

5.2.2.3 Solution for Coolant Temperatures

In solving for the temperatures of material points, the temperature variation of coolant points over the time step is neglected. The coolant point temperatures remain constant while the solution for the material points is performed. The coolant point temperatures are then calculated to satisfy a heat balance with the material points adjacent to the coolant block boundaries. Particular attention must be given to choosing appropriately small time steps when coolant temperatures are expected to change rapidly.

The error in the solution for coolant temperatures may be intensified when there is radiation across an internal coolant block. The heat transported by radiation at an internal coolant block boundary is accounted for in the value used for the overall conductances adjacent to that boundary.

Whenever radiation is present, the unit value of the overall conductance between a material point and an internal coolant point contains a component

$$h_r = \frac{\mathfrak{F}_{1-2} (T_{s1}^4 - T_{s2}^4)}{T_1 - T_{c1}}$$

where

h_r is the component of overall unit conductance representing radiation,
Btu/hr-ft²-°R

T_1 is the temperature of the material point, °R

T_{c1} is the temperature of the internal coolant point adjacent to the
material point, °R

T_{s1} is the local temperature of the internal coolant block boundary
separating the material point at T_1 and the coolant point at T_{c1} , °R

T_{s2} is the local temperature of the internal coolant block boundary
directly opposite the location of T_{s1} , °R

$\mathfrak{F}_{1-2}$ is the overall interchange factor for one-dimensional radiation
between the temperatures T_{s1} and T_{s2} , dimensionless

Within a given iteration, the T_1 used to calculate h_r is the value from the preceding iteration, whereas the T_1 used to calculate coolant temperatures is the value obtained from the current iteration. The result is that when coolant temperatures are calculated there is an inconsistency between h_r and the value of $T_1 - T_{c1}$ on which h_r is based. In calculations where an internal coolant temperature closely approaches the adjacent material temperature, h_r becomes large and also sensitive to very small changes in $T_1 - T_c$. Under this condition, an unusually large error may be incurred in the calculation of coolant temperatures. In problems where internal coolant temperatures approach adjacent material temperatures, particular attention should be given to eliminating any dependence of transient results upon the size of the time step.

The above error in coolant temperatures may become so large that the calculation is terminated due to some condition such as a floating point overflow. So that all such problems may be run to completion, the code

includes the following checks and arbitrary corrective measures which are applied in calculating overall conductances and coolant temperatures for the case of radiation across an internal coolant block:

1. All values of $T_1 - T_{c1}$ are checked before overall conductances between materials and internal coolants are calculated. At locations where $|T_1 - T_{c1}| < 0.10^\circ\text{R}$ on either boundary of the coolant block, all heat transfer by radiation is excluded for the iteration being performed.
2. The exit temperature calculated for each coolant node is checked to determine if it lies within limits dictated by the inlet temperature and the adjacent material point temperatures. If it is outside these limits, the point (i.e., the mean) and the exit temperatures for the node are set equal to the inlet temperature. In other words, if an error is detected, the coolant node is treated as being adiabatic for the current iteration. This is done so that coolant temperature errors will not be propagated to adjacent coolant nodes.

Unless the steady-state option is being used, one of the error messages given in Appendix A under Subroutines CONDOC and COOL is printed whenever either of the above corrective measures is applied. If the steady-state option is being used, the messages are printed only when the corrective measures are applied during the smoothing iterations described in Section 2.4. The messages give the times and locations where corrective measures were applied so that if they are printed some judgement as to the validity of results can be made. If results are not considered satisfactory, the corrective measures can usually be eliminated from the solution by rerunning the problem with smaller time steps.

5.3 ONE-DIMENSIONAL TREATMENT OF RADIATION

The radiation across gaps and internal coolant blocks is calculated under assumptions consistent with one-dimensional radiation across a narrow space. These assumptions are:

1. Radiation occurs only in a direction normal to the emitting surfaces.

2. The geometrical view factor is 1.0.
3. The mean effective area for heat transfer (both radiation and conduction) across gaps in the radial direction may be defined as that at the high radial boundary of the gap.

Errors due to the above assumptions can become significant for radiation across internal coolant blocks and gaps which are not narrow.

5.4 ROUND OFF ERRORS

Single precision is used throughout the code. Therefore, only eight significant figures are available. Situations can develop where temperature differences in the heat balance equations are so small that they appear only beyond the eighth significant figure. Changes in point temperatures between successive iterations are then lost in the roundoff. In running a problem under these conditions an apparent convergence will be obtained, but if further iterations are performed using a smaller time step, the solution will change.

Probably the most common situation involving roundoff error arises in problems where the thermal conductivities of some materials are very high. The heat balance contains terms involving products of conductances and point temperature differences. If the conductances are large, due to large thermal conductivities, the true temperature difference components of these terms can become so small as to appear only beyond the eighth figure.

6. TECHNIQUES

Many of the apparent limitations of the program can be overcome through techniques such as those described in the following sections and in Ref. 2. These techniques involve the use of materials having thermal parameters which are chosen to impose some desired condition on a problem. Such materials are called dummies because they are not physically present in the system being modeled.

Dummy materials are often specified with extreme (i.e., very high or very low) values of one or more thermal parameters. For dummy solid materials it is important to consider the effect of extreme values on the maximum stable time step. As indicated by the proportionality in Section 5.1.1, high thermal conductivity and/or low volume specific heat tend to reduce the maximum stable time step at any point where they are applied. In using dummy materials with such extreme parameters, it should be insured that they are no more extreme than required to impose the desired conditions.

6.1 BOUNDARY CONDITIONS

All boundary conditions must be described in terms of thermal parameters of coolants contained in the external blocks. Some special cases are illustrated below.

6.1.1 Adiabatic Boundary

A coolant having very low heat transfer coefficient is required. The other coolant parameters are not important. The standard coolant function values

$$\begin{aligned}H-A(X) &= 10^{-6} \text{ Btu/hr-ft}^2\text{-}^\circ\text{R} \\TIN-A(X) &= 460.^\circ\text{R} \\FLO-A(X) &= 10^6 \text{ lb/hr} \\SPH-(X) &= 1. \text{ Btu/lb-}^\circ\text{R}\end{aligned}$$

are usually adequate for defining an adiabatic boundary. As discussed in Section 2.5.3, these values will automatically be used if definitions of these functions are omitted in the thermal parameter function input.

6.1.2 Constant Temperature Boundary

The coolant must be isothermal at the desired boundary temperature and must also have a very high heat transfer coefficient.

$$H-A(X) = \text{very high}$$

$$TIN-A(X) = \text{desired boundary temperature}$$

$$[FLO-A(X)] \times [SPH-(X)] = \text{very high}$$

6.1.3 Boundary With Radiation

Radiation at an external boundary can be modeled by using a dummy coolant whose heat transfer coefficient is a function of surface temperature ST. As an example, consider a case where it is desired to transfer heat at an external boundary by combined convection and radiation. Convection between the local boundary and bulk coolant temperatures T_w and T_c , respectively, is described by a film coefficient h_c . There is radiant heat exchange with some surface which is in thermal equilibrium with the coolant. This radiation is one-dimensional and is described by an interchange factor $\mathfrak{F}$ and the Stefan-Boltzmann constant σ . The combined heat transfer is described by an effective overall heat transfer coefficient h_e , which is

$$h_e = \frac{\mathfrak{F}\sigma(T_w^4 - T_c^4)}{(T_w - T_c)} + h_c$$

If

$$h_c = 10. \text{ Btu/hr-ft}^2\text{-}^\circ\text{R}$$

$$\mathfrak{F} = 0.50$$

$$\sigma = 0.1713 \times 10^{-8} \text{ Btu/hr-ft}^2\text{-}^\circ\text{R}^4$$

then the combined heat transfer can be described by the following function for the heat transfer coefficient of the coolant:

$$H-A(X) = .50*.1713E-8*(ST**4-DR**4)/(ST-DR)+10.$$

where ST is the local surface temperature and DR is the local coolant temperature, both in °R. If there is no convection, the term representing h_c is not included. The external source-sink temperature need not be the local coolant temperature. It may be a constant or it may be defined by an additional arithmetic statement function using any of the variables which are allowed in H-A(X). Either of these would be substituted for DR in the above function.

6.2 IRREGULAR EXTERNAL BOUNDARIES

As discussed in Section 2.1.3, each of the four external boundaries must coincide continuously with one of the six extreme grid planes. This limitation can be overcome to some extent by using dummy materials in spaces between the external boundaries dictated by extreme grid planes and the true external boundaries of the problem. The thermal parameters of these dummy materials can be selected to impose the desired boundary conditions.

Examples 1 and 2 of this technique are given in Figs. 5 and 6, respectively. The illustrations in these figures are two-dimensional and should be considered as cross-sections through three-dimensional systems. In Example 1, a dummy material is used to simulate the constant temperature coolant and gaps are used to simulate the heat transfer coefficients. The dummy material is given a high specific heat and thermal conductivity so that all points which it contains remain at their initial temperatures. These initial temperatures are specified as the constant coolant temperature T_c .

Example 2 illustrates the use of internal coolants to impose desired conditions on an external boundary which has, of necessity, been compromised from its true configuration. A partial compensation for the boundary distortion is made by applying the heat transfer area correction to h as indicated in Fig. 6.

Dummy materials could be used in Example 2; however, the use of dummy internal coolants in Example 1 would require an unnecessary compromise of the true boundary conditions. To fill the dummy space of Example 1 with coolant blocks would require that some part of either the h_1 or the h_2

DUMMY THERMAL PARAMETERS

MATERIALS

RCON1(X) = HIGH, ACON1(X) = HIGH, SPEC1(X) = VERY HIGH
 INITIAL TEMPERATURE OF 1 = T_c

GASES

$GCON1(X) = h_1 t_1$
 $GCON2(X) = h_2 t_2$

NO RADIATION: ALL EMISSIVITIES = 0

COOLANTS

COOLANT NUMBER	H _ A(X)	TIN _ A(X)	FLO _ A(X)	SPH _ (X)
1	10^{-6}	460	10^6	1.0
2	h_3	T_c	VERY HIGH	1.0

COOLANT 1 IS AN ADIABATIC BOUNDARY COOLANT AS
 DESCRIBED IN 6.1.1.

EXCEPT FOR FLOW DIRECTION COOLANT 2 IS NOT
 ACTUALLY A DUMMY.

Fig. 5. Irregular external boundaries, Example 1

TRUE CONFIGURATION

MODEL FOR TAC3D INPUT

C-n = COOLANT n

DUMMY MATERIAL THERMAL PARAMETERS

COOLANT NUMBER	H _ A(X)	TIN _ A(X)	FLO _ A(X)	SPH _ (X)
1	$\left(h\right)\left(\frac{\ell_1}{\ell_1'}\right)$	T_c	VERY HIGH	1.0
2	$\left(h\right)\left(\frac{\ell_2}{\ell_2'}\right)$	T_c	VERY HIGH	1.0
3	$\left(h\right)\left(\frac{\ell_3}{\ell_3'}\right)$	T_c	VERY HIGH	1.0
4	10^{-6}	460.0	10^6	1.0

COOLANT 4 IS AN ADIABATIC BOUNDARY COOLANT
AS DESCRIBED IN 6.1.1

Fig. 6. Irregular external boundaries, Example 2

surface be adiabatic. This is because coolant block heat transfer occurs only normal to the flow direction, as illustrated in Item d of Fig. 4.

If on a particular problem there is a choice of using either dummy material blocks or dummy internal coolants, that of using internal coolants is preferable. The principal reason is that the high thermal conductivity required for the dummy material blocks can cause stability problems and round-off errors as discussed in Chapter 5.

6.3 INTERNAL DUMMY MATERIALS

Dummy material and coolant techniques can be applied in describing special internal conditions for a problem. Examples are internal sources and sinks or internal voids. These can be modeled by dummy materials or coolants having extreme values of the appropriate thermal parameters.

APPENDIX A

PROGRAM ERROR MESSAGES

```

=====
SUBROUTINE BOUNDA
=====
    BOUNDA1  THE LOW BOUNDARY FOR A BLOCK IS TOO LARGE OR
              LARGER THAN THE LARGEST BOUNDARY DEFINED IN THAT
              DIMENSION.
    BOUNDA2  THE HIGH BOUNDARY OF A BLOCK IS LARGER THAN THE
              LARGEST BOUNDARY DEFINED IN THAT DIMENSION.
    BOUNDA3  AN INTERNAL COOLANT BLOCK HAS BEEN SPECIFIED IN
              VIOLATION OF CODE LIMITATIONS. MOST LIKELY ERRORS ARE
              1.  A NORMAL COOLANT HAS BEEN SPECIFIED FOR A
                  PROBLEM IN CYLINDRICAL COORDINATES.
              2.  AN INTERNAL COOLANT HAS TWO BOUNDARIES WHICH
                  ARE COINCIDENT.
=====
SUBROUTINE CHECK
=====
    CHECK1   THERE ARE TOO MANY RADIAL POINTS AND GRID LINES.
    CHECK2   THERE ARE TOO MANY AXIAL POINTS AND GRID LINES.
    CHECK3   THERE ARE TOO MANY THETA POINTS AND GRID LINES.
    CHECK4   THE RADIAL GRID DATA IS OUT OF ORDER.
    CHECK5   THE AXIAL GRID DATA IS OUT OF ORDER.
    CHECK6   THE THETA GRID DATA IS OUT OF ORDER.
    CHECK7   THERE ARE TOO MANY BLOCKS.
    CHECK8   THE LOW RADIAL BOUNDARY IS LARGER THAN THE HIGH
              RADIAL BOUNDARY FOR SOME BLOCK.
    CHECK9   THE LOW AXIAL BOUNDARY IS GREATER THAN THE
              HIGH AXIAL BOUNDARY FOR SOME BLOCK.
    CHECK10  THE LOW THETA BOUNDARY IS GREATER THAN THE
              HIGH THETA BOUNDARY FOR SOME BLOCK.
    CHECK11  THE MATERIAL NUMBER FOR A BLOCK IS LARGER THAN THE
              MAXIMUM NUMBER OF MATERIALS AND COOLANTS ALLOWED.
    CHECK12  A RADIAL GAP MATERIAL NUMBER IS TOO HIGH.
    CHECK13  AN AXIAL GAP MATERIAL NUMBER IS TOO HIGH.
    CHECK14  AN THETA GAP MATERIAL NUMBER IS TOO HIGH.
=====
SUBROUTINE CONDOC
=====
    CONDOC1 , CONDOC2 , CONDOC3
              THESE MESSAGES INDICATE THAT RADIATION BETWEEN THE
              BOUNDARIES OF AN INTERNAL COOLANT BLOCK WAS NEGLECTED
              IN ORDER TO COMPLETE AN ITERATION. A SMALLER TIME

```

```

STEP SHOULD BE USED IF THE NEGLECT OF RADIATION IS
SIGNIFICANT.
CONDUCT1  RADIATION WAS NEGLECTED BETWEEN RADIAL GAPLINES IGL
AND IGH AT AXIAL POINT LEVEL J, THETA POINT LEVEL K.
CONDUCT2  RADIATION WAS NEGLECTED BETWEEN AXIAL GAPLINES JGL
AND JGH AT RADIAL POINT LEVEL I, THETA POINT LEVEL K.
CONDUCT3  RADIATION WAS NEGLECTED BETWEEN THETA GAPLINES KGL
AND KGH AT RADIAL POINT LEVEL I, AXIAL POINT LEVEL J.
=====
SUBROUTINE COOL
=====
COOL1 , COOL2 , COOL3
THERE IS A PROGRAM LIMITATION WHICH MAY BE
ENCOUNTERED WHEN CALCULATING COOLANT TEMPERATURES IN
THE PRESENCE OF RADIATION BETWEEN COOLANT BLOCK
BOUNDARIES. IT ARISES WHEN A COOLANT POINT AND AN
ADJACENT MATERIAL POINT APPROACH THE SAME
TEMPERATURE. THESE MESSAGES INDICATE THAT THIS
LIMITATION HAS BEEN ENCOUNTERED AND CIRCUMVENTED
BY ASSIGNING ZERO COOLANT HEAT TRANSFER AT THE
LEVEL INDICATED. BY USING A SMALLER TIME STEP,
THIS DIFFICULTY CAN USUALLY BE ELIMINATED.
COOL1  HEAT TRANSFER WITHIN A RADIAL FLOW COOLANT IS
ZERO IN BLOCK L AT RADIAL POINT LEVEL I.
COOL2  HEAT TRANSFER WITHIN AN AXIAL FLOW COOLANT IS
ZERO IN BLOCK L AT AXIAL POINT LEVEL J.
COOL3  HEAT TRANSFER WITHIN A THETA FLOW COOLANT IS
ZERO IN BLOCK L AT THETA POINT LEVEL K.
=====
SUBROUTINE FLUDAT
=====
FLUDAT1  THE INDEPENDENT VALUE LIES OUTSIDE THE FLOW RATE
FUNCTION RANGES.
FLUDAT2  THE INDEPENDENT VALUE LIES OUTSIDE THE INLET
TEMPERATURE FUNCTION RANGES.
FLUDAT3  THE REYNOLDS NUMBER LIES OUTSIDE THE SPECIFIED RANGES
=====
SUBROUTINE FLOWCA
=====
FLOWCA1  THE REYNOLDS NUMBER LIMITS ARE NOT IN SEQUENCE.
FLOWCA2  THE FLOW RATE LIMITS ARE NOT IN SEQUENCE.
FLOWCA3  THE INLET TEMPERATURE LIMITS ARE NOT IN SEQUENCE.
=====
SUBROUTINE GLOME1
=====
GLOME11  SOME POINT HAS A NEGATIVE CALCULATED VOLUME.
=====
SUBROUTINE GRID
=====
GRID1    THERE ARE MORE GRIDLINES THEN ALLOWED IN ONE OF
THE DIMENSIONS. A BLANK CARD HAS BEEN LEFT OUT.

```

```

=====
SUBROUTINE INITEM
=====
    INITEM1  THE TEMPERATURE BLOCK'S LOWER RADIAL BOUNDARY DOES
              NOT COINCIDE WITH ANY OF THE RADIAL GRID
              BOUNDARIES.
    INITEM2  THE TEMPERATURE BLOCK'S UPPER RADIAL BOUNDARY DOES
              NOT COINCIDE WITH ANY OF THE RADIAL GRID
              BOUNDARIES.
    INITEM3  THE TEMPERATURE BLOCK'S LOWER AXIAL BOUNDARY DOES
              NOT COINCIDE WITH ANY OF THE AXIAL GRID BOUNDARIES.
    INITEM4  THE TEMPERATURE BLOCK'S UPPER AXIAL BOUNDARY DOES
              NOT COINCIDE WITH ANY OF THE AXIAL GRID BOUNDARIES.
    INITEM5  THE TEMPERATURE BLOCK'S LOWER THETA BOUNDARY DOES
              NOT COINCIDE WITH ANY OF THE THETA GRID BOUNDARIES.
    INITEM6  THE TEMPERATURE BLOCK'S UPPER THETA BOUNDARY DOES
              NOT COINCIDE WITH ANY OF THE THETA GRID BOUNDARIES.
    INITEM7  AN INITIAL TEMPERATURE HAS NOT BEEN ASSIGNED TO SOME
              INTERNAL POINT.
=====
SUBROUTINE INPUT
=====
    INPUT1   THE GEOMETRY TYPE DESIRED HAS BEEN MISSPELLED.
    INPUT2   THE LOW RADIAL BLOCK BOUNDARY OF SOME BLOCK DOES
              NOT COINCIDE WITH A RADIAL GRID LINE.
    INPUT3   THE HIGH RADIAL BLOCK BOUNDARY OF SOME BLOCK DOES
              NOT COINCIDE WITH A RADIAL GRID LINE.
    INPUT4   THE LOW AXIAL BLOCK BOUNDARY OF SOME BLOCK
              DOES NOT COINCIDE WITH AN AXIAL GRID LINE.
    INPUT5   THE HIGH AXIAL BLOCK BOUNDARY OF SOME BLOCK
              DOES NOT COINCIDE WITH AN AXIAL GRID LINE.
    INPUT6   THE LOW THETA BLOCK BOUNDARY OF SOME BLOCK
              DOES NOT COINCIDE WITH A THETA GRID LINE.
    INPUT7   THE HIGH THETA BLOCK BOUNDARY OF SOME BLOCK
              DOES NOT COINCIDE WITH A THETA GRID LINE.
=====
SUBROUTINE MP2
=====
    MP2  1    THE CURRENT TIME OF THE PREVIOUSLY PUNCHED
              TEMPERATURE DISTRIBUTION IS GREATER THAN THE ENDING
              TIME OF ANY GIVEN TIME STEP.
=====
SUBROUTINE POINTS
=====
    POINTS1  A PART OF THE SYSTEM WAS NOT DESCRIBED BY ANY BLOCK.
    POINTS2  THERE ARE TOO MANY RADIAL GAP LINES.
    POINTS3  THERE ARE TOO MANY AXIAL GAP LINES.
    POINTS4  THERE ARE TOO MANY THETA GAP LINES.
    POINTS5  A PART OF THE SYSTEM HAS BEEN DESCRIBED BY MORE THAN
              ONE BLOCK.
    POINTS6  NO FLOW DIRECTION HAS BEEN ASSIGNED FOR SOME COOLANT.

```

POINTS7 1. AN EXTERNAL COOLANT IS FLOWING-INTO A RADIAL BOUNDARY,OR
 2. AN INTERNAL RADIAL FLOW COOLANT BLOCK IS NOT TRAVERSED BY AT LEAST ONE RADIAL GRID LINE.

POINTS8 1. AN EXTERNAL COOLANT IS FLOWING INTO AN AXIAL BOUNDARY,OR
 2. AN INTERNAL AXIAL FLOW COOLANT BLOCK IS NOT TRAVERSED BY AT LEAST ONE AXIAL-Z GRID LINE.

POINTS9 1. AN EXTERNAL COOLANT IS FLOWING INTO A THETA BOUNDARY,OR
 2. AN INTERNAL THETA FLOW COOLANT BLOCK IS NOT TRAVERSED BY AT LEAST ONE THETA GRID LINE.

POINTS10 A GAP HAS BEEN SPECIFIED ON THE HIGH RADIAL BOUNDARY OF A COOLANT.

POINTS11 A GAP HAS BEEN SPECIFIED ON THE LOW RADIAL BOUNDARY OF A COOLANT.

POINTS12 A GAP HAS BEEN SPECIFIED ON THE HIGH AXIAL BOUNDARY OF A COOLANT.

POINTS13 A GAP HAS BEEN SPECIFIED ON THE LOW AXIAL BOUNDARY OF A COOLANT.

POINTS14 A GAP HAS BEEN SPECIFIED ON THE HIGH THETA BOUNDARY OF A COOLANT.

POINTS15 A GAP HAS BEEN SPECIFIED ON THE LOW THETA BOUNDARY OF A COOLANT.

=====

SUBROUTINE PRETEM

=====

PRETEM1 THE PROBLEM SIZE DOES NOT MATCH THE INITIAL TEMPERATURE DISTRIBUTION DATA.

=====

SUBROUTINE SURT

=====

SURT 1 INSTABILITY ENCOUNTERED WHILE CALCULATING THE RADIAL BOUNDARY TEMPERATURES OF A GAP. USE A SMALLER TIME STEP.

SURT 2 INSTABILITY ENCOUNTERED WHILE CALCULATING THE AXIAL BOUNDARY TEMPERATURES OF A GAP. USE A SMALLER TIME STEP.

SURT 3 INSTABILITY ENCOUNTERED WHILE CALCULATING THE THETA BOUNDARY TEMPERATURES OF A GAP. USE A SMALLER TIME STEP.

=====

SUBROUTINE TIME

=====

TIME1 THE FIRST TIME CARD DOES NOT SPECIFY TIME UNITS. THE PROBLEM IS NOT AFFECTED IF THE STEADY STATE OPTION IS BEING USED.

TIME2 TOO MANY TIME HISTORY CARDS HAVE BEEN READ.

TIME3 THE INPUT CONTAINS NO TIME HISTORY CARDS AND THE STEADY STATE OPTION IS NOT BEING USED.

APPENDIX B

EXAMPLE USE OF ADDITIONAL ARITHMETIC STATEMENT FUNCTIONS

A typical use of additional functions would be to calculate the Reynolds numbers and heat transfer coefficients of coolants with properties evaluated at the mean film temperature. An example of this is given below where the additional functions $TFILM(X)$, $CON(X)$, $VIS(X)$, $CP(X)$ and $PR(X)$ are used to define the coolant parameter functions $REYN2(X)$, $REYN7(X)$, $H2A(X)$ and $H7A(X)$.

```
TFILM(X) = (ST+DR)/2.  
CON(X) = 1.29E-3*TFILM(X)**.674  
VIS(X) = 6.9E4*TFILM(X)**.674  
CP(X) = 1.242  
PR(X) = CP(X)*VIS(X)/CON(X)  
REYN2(X) = 4.*FR/ (.524*VIS(X))  
REYN7(X) = 4.*FR/ (1.048*VIS(X))  
H2A(X) = .021*(RE**.8)*(PR(X)**.4)*(CON(X)/.167)  
H7A(X) = .021*(RE**.8)*(PR(X)**.4)*(CON(X)/.333)
```

NOTES:

1. The variables used in the additional functions must be only those which are allowed for the thermal parameter functions in which they are used. This usage is defined in Section 2.5.2.
2. One additional function may be used in several different thermal parameter functions provided that in the sequences of statements inserted into MADATA and/or FLODAT its definition always precedes its use.

.

.

.

.

APPENDIX C

EXAMPLE USE OF FUNCTION SUBPROGRAMS

It is desired that the specific heat of Material 3 be defined in two different temperature ranges:

$$\text{SPEC3}(X) = 490. * (6.364\text{E-}5 * (\text{DR} - 460.) + .107) \text{ for } \text{DR} < 960^{\circ}\text{R}$$

$$\text{SPEC3}(X) = 490. * (3.333\text{E-}5 * (\text{DR} - 460.) + .119) \text{ for } \text{DR} \geq 960^{\circ}\text{R}$$

This will be done using a function subprogram named VSH.

In the material parameter function statements assign:

$$\text{SPEC3}(X) = \text{VSH}(\text{DR})$$

At the location shown in Section 4.2 for function subprograms, place the following cards:

```
VP HDG      any alphanumeric heading in Cols. 13-72      (Note 1)
VI      FOR      VSH/S,VSH/S,VSH/R                        (Note 1)
      FUNCTION VSH(TEM)
      IF(TEM.GE.960.)GO TO 10
      VSH = 490.*(6.364E-5*(TEM-460.)+.107)
      RETURN
10 VSH = 490.*(3.333E-5*(TEM-460.)+.119)
      RETURN
      END
```

NOTES:

1. The first two lines in the above example are control cards for the Univac 1108 computer.
2. There may be more than one variable in the function calling sequence. An example is a thermal

parameter function

$$\text{HEAT9}(X) = \text{HGF}(\text{DR}, \text{FTZ})$$

which could be evaluated from the function subprogram labeled

FUNCTION HGF(TEM,Z) .

In the Fortran programming of HGF, TEM corresponds to DR and Z corresponds to FTZ.

3. A function subprogram name may be used like an additional arithmetic statement function in that it may be used in several different parameter functions. For example,

$$\text{SPEC3}(X) = \text{VSH}(\text{DR})$$

$$\text{SPEC6}(X) = 1.10 * \text{VSH}(\text{DR})$$

4. The variables in the function subprogram's calling sequence must be only those which are allowed for the thermal parameter functions in which they are used.

APPENDIX D

DIMENSIONAL LIMITS

Computer core capacity available for array storage is distributed according to the values assigned to the following parameters:

IQ - number of X (radial) points
JQ - number of Y (axial) points
KQ - number of Z (theta) points
IGQ - number of X (radial) gap lines
JGQ - number of Y (axial) gap lines
KGQ - number of Z (theta) gap lines
LQ - number of blocks

In the standard version of TAC3D the values are:

$$IQ = 12, JQ = 12, KQ = 12$$

$$IGQ = 4, JGQ = 4, KGQ = 4$$

$$LQ = 50$$

The values of the above parameters are defined in each subroutine by a single FORTRAN statement. It is relatively simple to change these values and compile a special version of the code which has different dimensional limits than the standard version.

The constraint on changing the parameter values is

$$P1 + P2 + P3 + P4 + P5 \leq CLIM$$

where

$$P1 = 7 \times IGQ \times JQ \times KQ$$

$$P2 = 7 \times JGQ \times IQ \times KQ$$

$$P3 = 7 \times KGQ \times IQ \times JQ$$

$$P4 = 11 \times IQ \times JQ \times KQ$$

$$P5 = 7 \times LQ$$

and

CLIM = 31830 .

The above value of CLIM is applicable only to the Univac 1108 computer installation at Gulf General Atomic. It includes an allowance of 1000 octal core locations for storage of the arithmetic statement functions (Section 2.5) to be compiled into Subroutines MADATA and FLODAT by the code user.

APPENDIX E

DUMPING

Values of program variables at the termination of a computer run can be printed by means of a core dump. The most useful types of dumps are described below. Descriptions of dumps and dump cards are directly applicable only to the Univac 1108 computer installation at Gulf General Atomic.

E.1 FULL CORE DUMP

Sometimes a problem fails to run successfully on the computer and the cause is not apparent from error messages alone. Assistance will have to be sought from a person familiar with the program and the computer system on which it is being used. The following card* will give a dump on error with decimal conversion of the core configuration at the time of the error.

VBREL PMD

The output pages required for dumps are not included in the page count which is subject to the specified maximum number of output pages.

E.2 DUMP OF A GIVEN COMMON ELEMENT

If it is known which variables are of interest, it is possible to dump only the named common elements which contain those variables. One of the most frequent applications for such a limited dump is in the running of a time test case. Here it is expected that the problem will terminate on a system error with no output results for the current iteration. However, the only results needed are the time and number of current iteration. These variables, named CURTI and NITER, respectively, are contained in the common

*The dump cards discussed in this appendix are always placed on the back of the input data deck. The dump card is the last card.

element named MISCXX. The following card will give a dump of this element on error:

VRE PMD MISCXX

The dump is contained in less than one printed page.

APPENDIX F

OUTPUT DATA TAPE

The data contained in the names defined below is written on a tape whenever the tape option is specified in the input of a problem.

CURTI	Current time, hr
FLOW(N)	Flow rate of Coolant N, lb/hr
IGR(I)	Index of the gap plane at radial grid plane I
IH(L)	Highest radial point index in Block L
IL(L)	Lowest radial point index in Block L
ILEN	An integer which is always 1
IMAX	Total number of radial points
ISHPAE	Coordinate system indicator: 0 = cylindrical, 1 = rectangular
JGZ(J)	Index of the gap plane at axial grid plane J
JH(L)	Highest axial point index in Block L
JL(L)	Lowest axial point index in Block L
JMAX	Total number of axial points
KGT(K)	Index of the gap plane at theta grid plane K
KH(L)	Highest theta point index in Block L
KL(L)	Lowest theta point index in Block L
KMAX	Total number of theta points
LMAX	Total number of blocks (material and coolant)
MAXFLO	Maximum number of coolants (15)
MB(L)	Number of the material or coolant in Block L
NITER	Current iteration number
NRG	Total number of radial gap planes
NZG	Total number of axial gap planes
NTG	Total number of theta gap planes
RL(I)	Radial grid plane coordinate, ft

T(I,J,K)	Temperature at radial point level I, axial point level J, theta point level K, °R
TI(N)	Inlet temperature of Coolant N, °R
TL(K)	Theta grid plane coordinate, radians
TO(N)	Outlet temperature of Coolant N, °R
ZA(I)	An array of 12 Hollerith words which contains the information on the first problem title card
ZL(J)	Axial grid plane coordinate, ft

All the above names which begin with the letters I through N contain integers. All other names contain floating point numbers except ZA, which contains Hollerith words.

The data is written on Tape Unit 15 by executing the following write statements:

```

WRITE(15)      ISHAPE, ILEN, IMAX, JMAX, LMAX, MAXFLO, NRG, NZG, NTG,
1              (ZA(I),I=1,12),(RL(I),I=1,IMAX),(ZL(J),J=1,JMAX),
2              (TL(K),K=1,KMAX),(IGR(I),I=1,IMAX),(JGZ(J),J=1,JMAX),
3              KGT(K),K=1,KMAX),(IL(L),IH(L),JL(L),JH(L),KL(L),KH(L),
4              MB(L),L=1,LMAX)
WRITE(15)      IMAX,JMAX,KMAX,MAXFLO,
1              CURTI,NITER,
2              (FLOW(N),TI(N),TO(N),N=1,MAXFLO),
3              ((T(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)

```

The first statement is executed one time as soon as all the input data has been processed and the geometrical constants calculated. The second statement is executed each time basic output is printed.

Whenever the tape option is specified a data tape number XXXX must be assigned by means of the following card:

```

▽  ASG      15 = XXXX

```

This card should be placed immediately after the program tape assign card in the card sequence given in Section 4.2.

REFERENCES

1. Douglas, J., "Alternating Direction Methods for Three Space Variables," Numer. Math., 4, 41-63 (1962).
2. 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.
3. Petersen, J. F., "TAC2D, A General Purpose Two-Dimensional Heat Transfer Computer Code - User's Manual," USAEC Report, GA-8868, Gulf General Atomic, September, 1969.
4. Clark, S. S. and Petersen, J. F., "TAC2D, A General Purpose Two-Dimensional Heat Transfer Computer Code - Mathematical Formulations and Programmer's Guide," USAEC Report GA-9262, Gulf General Atomic, September, 1969.
5. Richtmyer, R. D., Difference Methods for Initial Value Problems, Interscience Publishers, 1957.