LOCKHEED MARTIN ENERGY RESEARCH LIBRARIES
3 4456 0514389 5

OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION • NUCLEAR DIVISION
for the
U. S. ATOMIC ENERGY COMMISSION

ORNL-TM-3324

cy. 141

ORTHIS, ORTHAT — TWO COMPUTER PROGRAMS FOR SOLVING TWO-DIMENSIONAL STEADY-STATE AND TRANSIENT HEAT CONDUCTION PROBLEMS

R. C. Durfee and C. W. Nestor, Jr.

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this document, send in name with document and the library will arrange a loan.

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

ORNL-TM-3324

Contract No. W-7405-eng-26

MATHEMATICS DIVISION

ORTHIS, ORTHAT — TWO COMPUTER PROGRAMS FOR SOLVING TWO-DIMENSIONAL STEADY-STATE AND TRANSIENT HEAT CONDUCTION PROBLEMS

R. C. Durfee and C. W. Nestor, Jr.

JULY 1971

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
operated by
UNION CARBIDE CORPORATION
for the
U. S. ATOMIC ENERGY COMMISSION

LOCKHEED MARTIN ENERGY RESEARCH LIBRARIES

3 4456 0514389 5

TABLE OF CONTENTS

Page

Abstract 1

1. Introduction 1

2. Mathematical Analysis 4

2.1 General Problem Description 4
2.2 Mesh Layout and Description 7
2.3 Numerical Technique for Steady-State Heat Conduction . . 13
2.4 Numerical Technique for Transient Heat Conduction . . . 18
2.5 Position, Time, and Temperature Dependent Functions . . 24
2.6 Approximation of Nodal Melting and Freezing 28

3. Input Description 30

3.1 General 30
3.2 Expressions 35
3.3 Units 40
3.4 ***HEADER*** Section 43
3.5 ***MESH*** Section 48
3.6 ***NAMES*** Section 51
3.7 ***BOUNDARY*** Section 52
3.8 ***MATERIALS*** Section 58
3.9 ***TEMPERATURES*** Section 65
3.10 ***TIME*** Section 69
3.11 Parametric Studies 72
3.12 Restart Facilities 75
3.13 Maximum Allowable Sizes 77

4. Output Description 78

4.1 General 78
4.2 Printout of Input Data 81
4.3 Printout of Mesh Picture 85
4.4 Printout of Iteration Summary 88
4.5 Printout of Temperature Array 93
4.6 Steady-State Plotting Output 99

Page

5. Possible Future Improvements 102

5.1 Dynamic Storage Allocation 102

5.2 Treatment of Irregular and Internal Boundaries 103

5.3 Treatment of Radiation at Internal and External Boundaries 104

5.4 Automatic Calculation of Time Steps for Transient 105

5.5 Evaluation of Temperature-Dependent Conductivities on Node Interfaces for Transient 106

5.6 Automatic Extrapolation for Steady-State 107

5.7 Complete Regional Heat Rebalancing for Steady-State . . . 108

5.8 Plotting of Temperatures for Transient 109

Appendix A. Nomenclature 110

Appendix B. Summary of Input Expressions 113

Appendix C. Table of Acceptable Units 117

Appendix D. Listing of Main Program for ORTHIS 118

Listing of Main Program for ORTHAT 125

Appendix E. Listing of Skeleton Subprogram for CONDF (Steady-State Only) 135

Listing of Skeleton Subprogram for CONDF (Transient Only) 138

Listing of Skeleton Subprogram for SHEATF (Transient Only) 139

Listing of Skeleton Subprogram for SMASSF (Transient Only) 140

Listing of Skeleton Subprogram for QF (Transient and Steady-State) 141

Listing of Skeleton Subprogram for TOBARF (Transient Only) 142

Listing of Skeleton Subprogram for ARATEF (Transient Only) 144

Appendix F. Listing of Skeleton Subprograms for Sample I/O.FUNCTIONs 150

Appendix G. Derivation of Difference Equation for $\partial T/\partial n$ 151

Page

Appendix H. Study of Steady-State Convergence 154

Appendix I. Control Cards 160

References 162

Addendum 167

ORTHIS, ORTHAT — TWO COMPUTER PROGRAMS FOR SOLVING TWO-DIMENSIONAL STEADY-STATE AND TRANSIENT HEAT CONDUCTION PROBLEMS

R. C. Durfee and C. W. Nestor, Jr.

ABSTRACT

The two programs, ORTHIS and ORTHAT,* are designed to solve steady-state and transient heat conduction problems, respectively, in two-dimensional geometries. Either Cartesian (x-y) or cylindrical (r-z, r-θ) coordinate systems may be used. Thermal properties, heat generation rates, and boundary conditions may be functions of position, time, or temperature. The first program uses the iterative method of successive overrelaxation (SOR) to solve the steady-state problem. In addition to overrelaxation it also uses a dominant-error-mode extrapolation procedure to increase the rate of convergence. The second program uses a modified alternating-direction implicit method (ADI) to solve the transient problem. The input for both programs has been designed in an easily usable free-form style.

1. INTRODUCTION

We have written two FORTRAN IV programs, ORTHIS and ORTHAT, to solve heat conduction problems in two-dimensional geometries. The first program solves the steady-state problem while the second solves the transient problem. Many features are common to both programs, and therefore no distinction will be made between the two except when necessary. The user

*Note: ORTHIS = Oak Ridge Transfer of Heat in Steady State
ORTHAT = Oak Ridge Transfer of Heat at Transient

has a choice of either the Cartesian (x-y) coordinate system or the cylindrical (r-z, r-θ) coordinate system. Thermal properties, heat-generation rates, and boundary conditions may be functions of position, time, or temperature as specified by the user. In the transient program we have included an approximate treatment of nodal melting and freezing.

The first program uses the iterative method of successive over-relaxation[1] (SOR) to solve the steady-state problem. In addition to overrelaxation it also uses a dominant-error-mode extrapolation procedure[2] to increase the rate of convergence. We are also planning to include some type of regional heat rebalancing[3,4] to further accelerate convergence. The second program uses a modified alternating-direction implicit method[1] (ADI) to solve the transient problem. We found that the normal alternating-direction method was unstable for very small nodes such as those which occur in air gaps. This instability was due to the explicit approximations used in evaluating the time integral. Therefore we modified the ADI method by removing this explicit dependence and thereby removed the instability. This modification required a double mesh sweep at each time step.

The major goal in the development of these two programs was simplicity from the user's point of view. One of the most tedious jobs in setting up a heat transfer problem for the computer involves the geometrical description of the body to be studied. The body must first be zoned into regions, the regions must be divided into some type of nodal lattice, materials must be assigned to the proper nodes, boundary conditions must be applied to the boundary nodes, and so on. We have attempted to minimize the amount of effort required to develop the geometrical

description and to prepare the corresponding input for the computer. In so doing we have designed an easily usable free-form style of input using keywords and symbols intermixed with the necessary numeric information. Another headache which is often encountered involves the readability of the computer output, especially the temperature distribution. To help solve this problem, we have developed routines for printing and plotting the temperatures in a map-like framework so that it is easy to see what the temperatures are anywhere over the whole mesh. In addition to temperature maps, the steady-state program can also plot temperature traverses. In the future we hope to increase the output capabilities of both codes.

We discuss the numerical techniques in detail in Section 2. The use of the codes is presented in Sections 3 and 4, illustrated by a sample problem. Finally, in Section 5, we point out some improvements which should be made in the codes.

2. MATHEMATICAL ANALYSIS

2.1 General Problem Description

The two programs, ORTHIS and ORTHAT, were originally developed for use in the analysis of heat transfer problems associated with the design of shipping casks for reactor fuel elements. Since the shipping casks were cylindrical in shape, these codes were designed primarily to handle cylindrical geometries. After these codes were written and used with shipping casks in the r-z geometry, we found that with a few modifications the x-y and r-θ geometries could be included. The internal composition of the assembly was assumed to be homogeneous within each of several regions, defined by the intersection of coaxial cylinders and planes perpendicular to the cylinder axis as shown in Fig. 2.1 below.

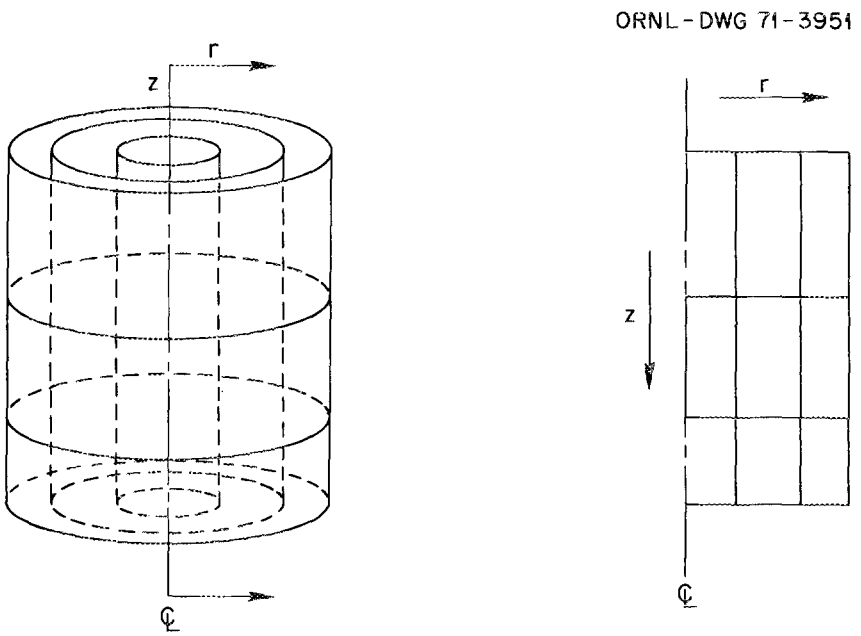


Figure 2.1

A different material may be placed within each region if desired. These regions are divided into nodes both vertically and horizontally as described in Section 2.2. Heat may flow from a given node to any of its four neighbors along paths which are parallel to the axes.

The center line of the assembly ($r = 0$) is always a symmetry axis, so the boundary conditions must be specified on the other three boundaries only. The boundary conditions may be specified by a fixed temperature, T_b, or a relation of the form

$$T_b + (1/a)(\partial T/\partial n)_b = T_{ext} \tag{2-1}$$

which can represent forced convection. When using Eq. (2-1) as a boundary condition, the user must specify the external temperature, T_{ext}, and the effective boundary layer thickness, $1/a$. This boundary layer thickness is simply the thermal conductivity of the surface material divided by the heat transfer coefficient, h. Of course, any part of the boundary may be insulated from the outside.

An initial temperature distribution must be used to begin a calculation. The distribution will probably be an educated guess in the case of a steady-state problem. However, in the case of a transient problem, this distribution will usually come from a previous steady-state solution.

Heat sources and sinks may be placed inside any node within the assembly. The user may wish to specify a regional heat source rather than a nodal heat source. In this case, the heat source at each node in the region is the total heat source divided by the volume of the region.

As stated previously, the codes will handle x-y and r-θ geometries as well as the r-z. Although the x-y geometry uses a spacing different

from the r-z, it looks the same since both are rectangular in shape. However, the mesh layout for the r-θ geometry is slightly different as described in Section 2.2. Since the three geometries are so similar, we will refer only to the r-z system unless some distinction is absolutely necessary.

2.2 Mesh Layout and Description

As was mentioned previously, the body to be studied is approximated by dividing it into regions, depending on its shape and material structure. These regions are formed by drawing lines parallel to the r and z axes as shown in Fig. 2.2 below. These lines must extend the complete length or width of the assembly. Each region may contain only one material (however, many regions may contain the same material). These regions are numbered horizontally (LH), vertically (LV), and internally (L) as shown. The user must specify the thickness (H) of each horizontal region and the thickness (h) of each vertical region.

ORNL-DWG 71-3952

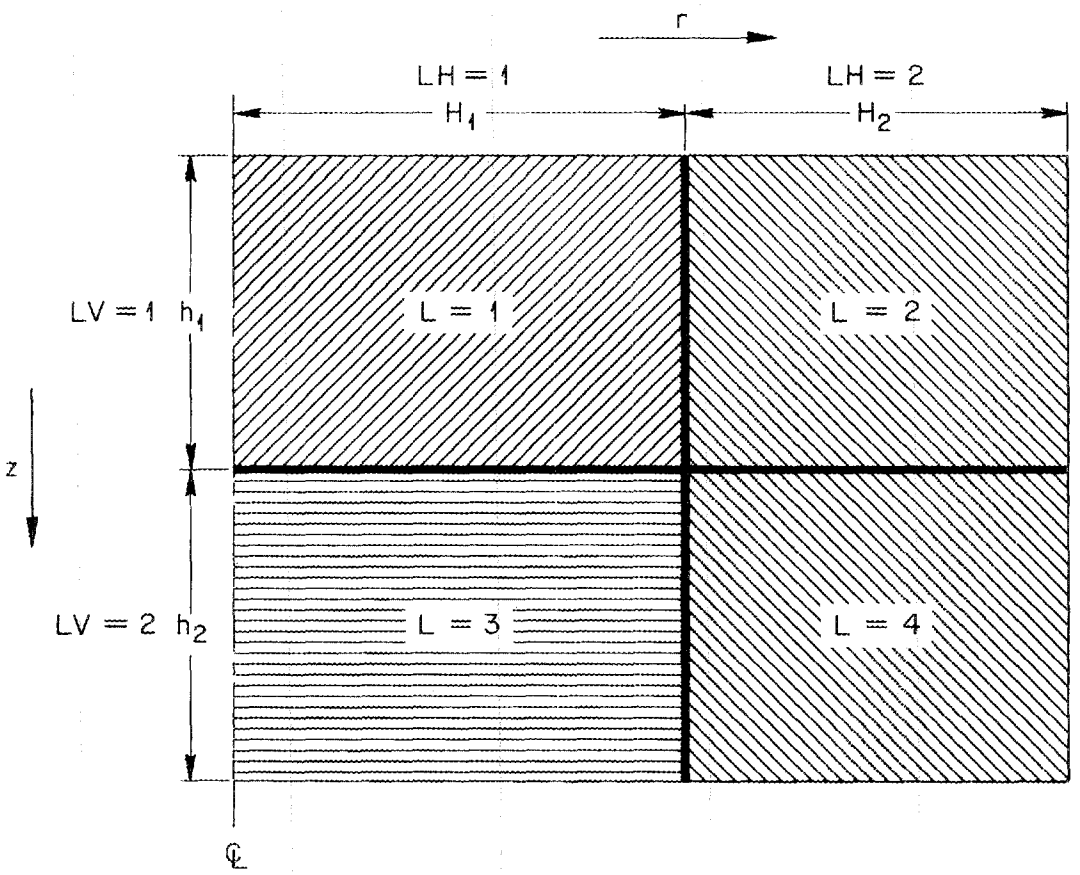


Figure 2.2

Once the regions have been constructed they must be divided into nodes both horizontally and vertically. This is done by specifying the number of nodes (n) in each horizontal region and the number of nodes (m) in each vertical region as shown in Fig. 2.3. The center of each node is called a nodal point, and it is at these nodal points that the temperatures are calculated. The nodes are also numbered horizontally (J), vertically (I), and internally (K) as shown below.

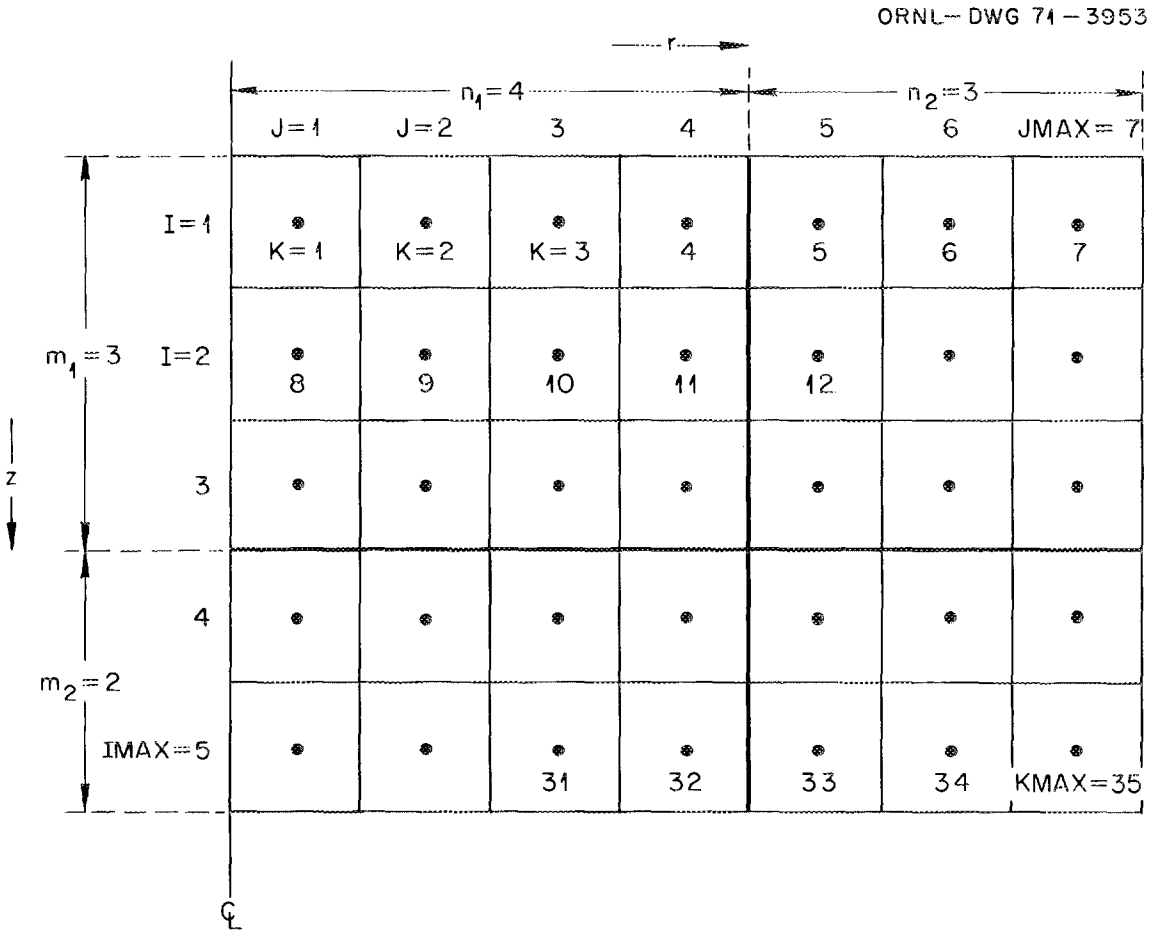


Figure 2.3

Although the user specifies the number of nodes in each region, he does not specify the actual position of the nodes or the nodal points. This is done by the program itself in such a way that every node within a given region has the same volume.[5] As a result of such a procedure in r-z and r-θ geometries, the nodal points are not equally spaced in the radial direction. However, the vertical intervals are equally spaced within a given region. Figure 2.4 and Eqs. (2-2) through (2-5) show the radial and vertical distances of the nodes and their nodal points.

ORNL-DWG 71-3954

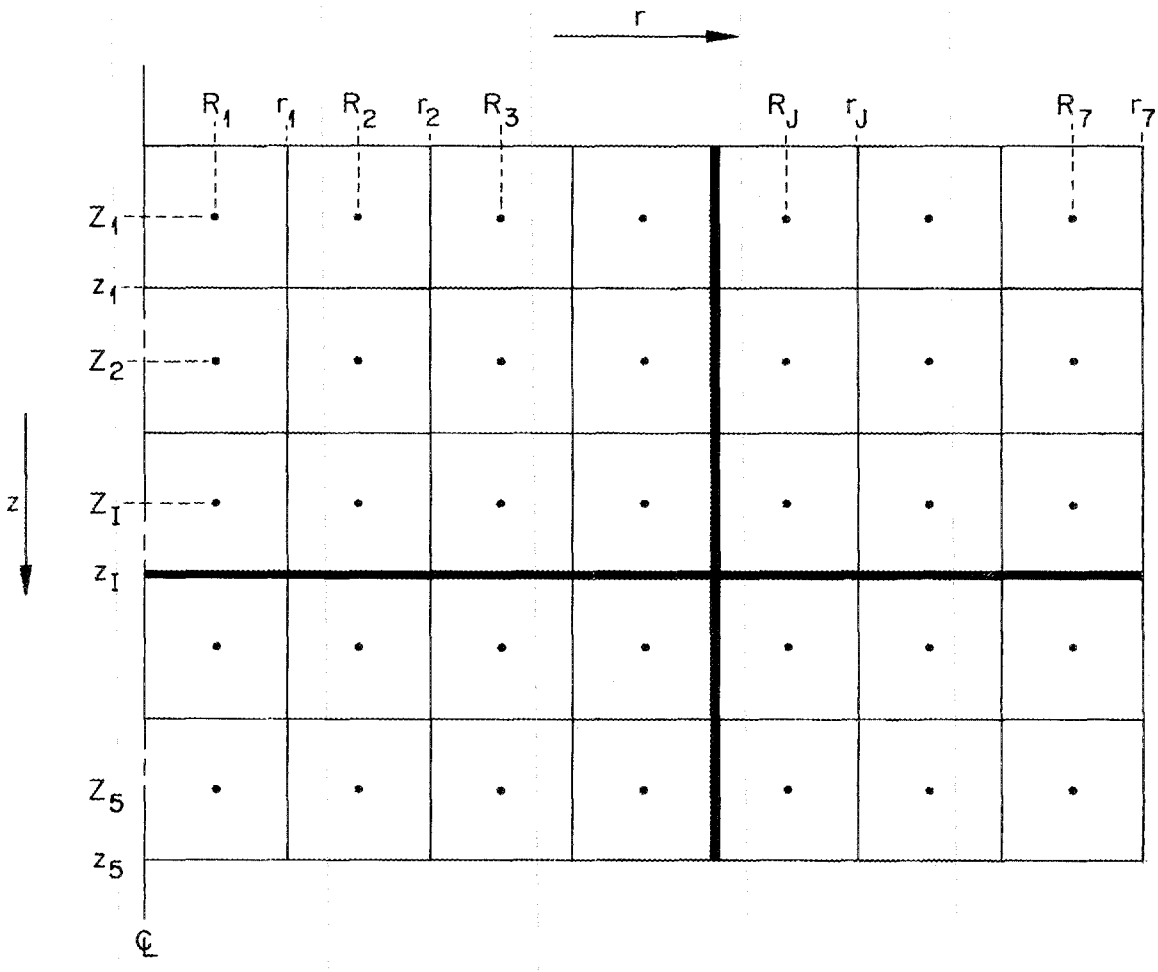


Figure 2.4

$$z_I = z_{I-1} + \frac{h_{LV}}{m_{LV}} \tag{2-2}$$

$$Z_I = \frac{z_I + z_{I-1}}{2} \tag{2-3}$$

$$r_J = \left[r_{J-1}^2 + \frac{(H_1 + H_2 + \ldots H_{LH})^2 - (H_1 + H_2 + \ldots H_{LH-1})^2}{n_{LH}} \right]^{\frac{1}{2}} \tag{2-4}$$

$$R_J = \frac{r_J + r_{J-1}}{2} \quad . \tag{2-5}$$

All of the previous mesh description for the r-z geometry also applies to the x-y geometry with one exception. The nodal points are now equally spaced in the horizontal direction. Thus the equations for the horizontal and vertical spacing are simply:

$$x_J = x_{J-1} + \frac{H_{LH}}{n_{LH}} \; ; \quad X_J = \frac{x_J + x_{J-1}}{2} \tag{2-6}$$

$$y_I = y_{I-1} + \frac{h_{LV}}{m_{LV}} \; ; \quad Y_I = \frac{y_I + y_{I-1}}{2} \quad . \tag{2-7}$$

When using the x-y geometry, boundary conditions must be specified for all four sides of the mesh.

In the case of the r-θ geometry, the equations for the horizontal and vertical spacing are identical to those in the r-z geometry. However, the mesh looks a lot different, as shown in Figure 2.5. As can be seen, the radial axis still represents the horizontal direction, but the vertical z axis has been replaced with an angular theta axis. The user might think of this r-θ mesh as being a deformed r-z mesh where the bottom has been stretched clockwise around the origin until it touches the top

(z = 0 line). Notice that the angular thicknesses are now given in units of radians rather than units of length such as feet or inches.

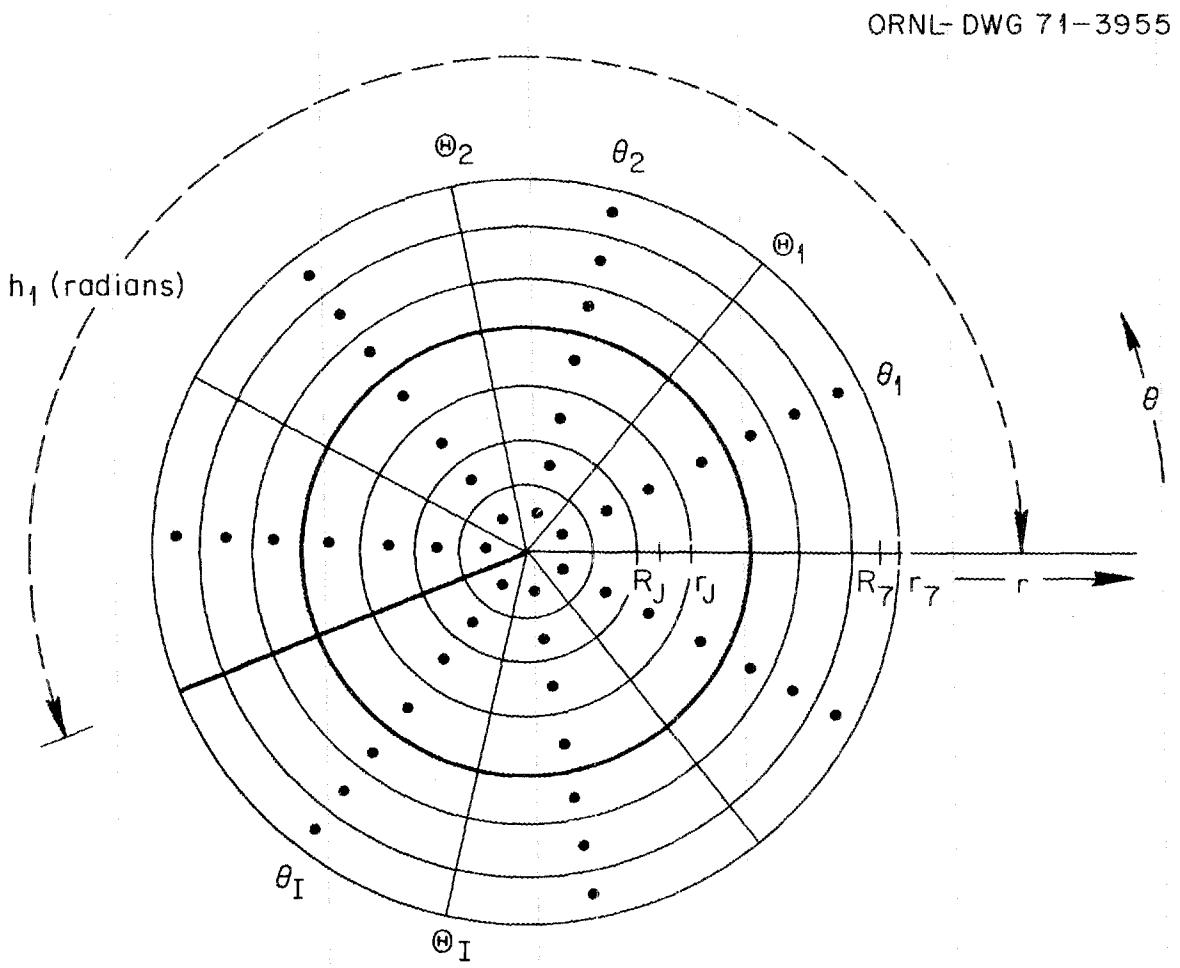


Figure 2.5

The equations for the r-θ spacing are

$$\theta_I = \theta_{I-1} + \frac{h_{LV}}{m_{LV}} ; \quad \Theta_I = \frac{\theta_I + \theta_{I-1}}{2} \tag{2-8}$$

$$r_J = \left[r_{J-1}^2 + \frac{(H_1 + \ldots H_{LH})^2 - (H_1 + \ldots H_{LH-1})^2}{n_{LH}} \right]^{\frac{1}{2}} ;$$

$$R_J = \frac{r_J + r_{J-1}}{2} . \tag{2-9}$$

When using the r-θ geometry, boundary conditions can be specified for only one side of the mesh (the outside edge), since the range of θ is from 0 to 2π in this version of the programs.

When laying out the mesh spacing, the user should try, within reason, to keep the same number of nodes per unit length in each region. This improves the accuracy of the calculations, especially at region boundaries where the internodal distances might be quite different on each side of the boundary.

2.3 Numerical Technique for Steady-State Heat Conduction

The partial differential equation which represents steady-state heat conduction in an isotropic material may be written in vector notation as

$$\vec{\nabla} \cdot k \vec{\nabla} T + q = 0 \tag{2-10}$$

where $\vec{\nabla}$ is the gradient operator, T is the temperature, k is the thermal conductivity, and q is the heat-generation rate per unit volume. Integrating over the nodal volume, V_K, we have

$$\int_{V_K} \vec{\nabla} \cdot k \vec{\nabla} T \, dV + \int_{V_K} q \, dV = 0 \quad . \tag{2-11}$$

Through the use of Green's first identity (or Gauss' theorem) we can rewrite this equation as

$$\int_{A_K} k \frac{\partial T}{\partial n} \, dA + \int_{V_K} q \, dV = 0 \tag{2-12}$$

where we are now integrating over the nodal surface, A_K. The $\frac{\partial T}{\partial n}$ represents the partial derivative of T with respect to the outward normal to the surface of the node, K. Assuming that k does not vary as we integrate over the surface area, and assuming that q does not vary as we integrate over the nodal volume we have

$$k_K \int_{A_K} \frac{\partial T}{\partial n} \, dA + q_K V_K = 0 \quad . \tag{2-13}$$

A thin portion of one of the nodes in the mesh is shown in Fig. 2.6. The four surfaces over which the surface integral is taken are numbered as shown. Assuming that $\frac{\partial T}{\partial n}$ does not vary over each surface of integration, we can write Eq. (2-13) as

$$k_K \sum_{S=1}^{4} A_{K,S} \frac{\partial T}{\partial n_{K,S}} + q_K V_K = 0 \tag{2-14}$$

ORNL-DWG 71-3956

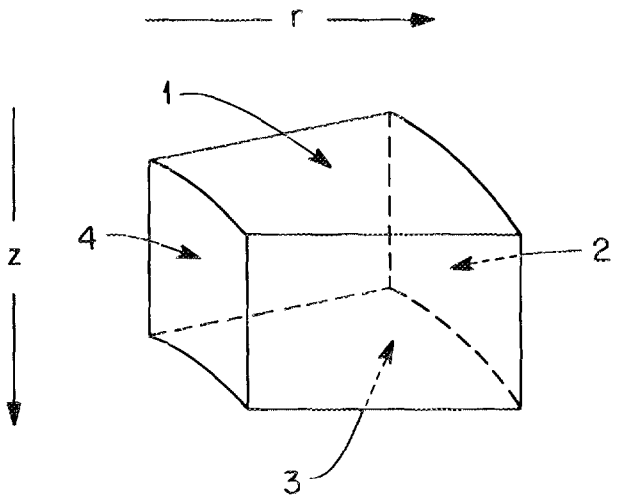


Figure 2.6

where $A_{K,S}$ is the area of side S, for node K, in the mesh. All that remains to be done now is to evaluate $\frac{\partial T}{\partial n_{K,S}}$ on each of the four sides of the node. This evaluation will depend upon whether a given side is touching (1) a node in the same region, (2) a node in a different region, or (3) the outside boundary. Regardless of which case we are considering, $\frac{\partial T}{\partial n_{K,s}}$ may be approximated by an equation of the form

$$\frac{\partial T}{\partial n_{K,S}} = \frac{\tau_{K,S} - T_K}{B_{K,S}} \tag{2-15}$$

where $\tau_{K,S}$ is a neighboring temperature, T_K is the temperature of the node itself, and $B_{K,S}$ is an internodal distance. $\tau_{K,S}$ and $B_{K,S}$ will depend upon the type of neighboring node (or boundary) touching each side of the given node, K. Since there are two types of boundary conditions allowed

(see Eq. (2-1)), there are four different equations to be chosen from in the evaluation of $B_{K,S}$. These four different cases are shown in Fig. 2.7 for the side, S = 2. Appendix G contains a derivation of Eq. (2-15).

By substituting Eq. (2-15) into Eq. (2-14) we now have

$$k_K \sum_{S=1}^{4} A_{K,S} \left(\frac{\tau_{K,S} - T_K}{B_{K,S}} \right) + q_K V_K = 0 \quad . \tag{2-16}$$

ORNL-DWG 71-3957

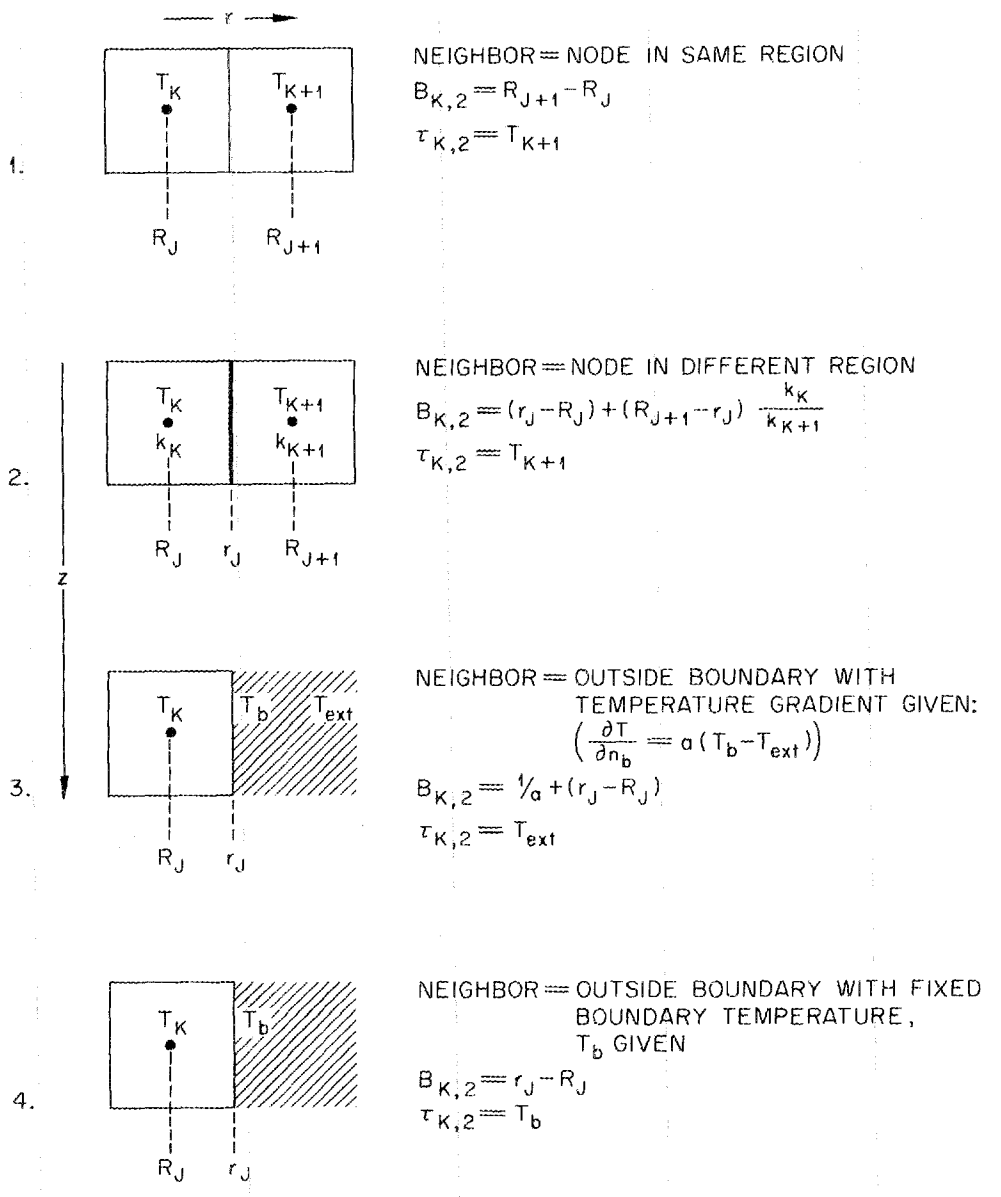


Figure 2.7

After rewriting this equation so that T_K appears on the left-hand side, we obtain our final equation for the temperature at node, K.

$$T_K = \frac{q_K V_K + k_K \sum_{S=1}^{4} \frac{A_{K,S} \tau_{K,S}}{B_{K,S}}}{k_K \sum_{S=1}^{4} \frac{A_{K,S}}{B_{K,S}}} \quad . \tag{2-17}$$

The temperature of node K is expressed in terms of $\tau_{K,S}$, the temperatures of its four surrounding neighbors. Since at the beginning of the steady-state problem we do not know any of the temperatures within the mesh, we must make an initial guess for all of them. An iterative procedure is then carried out in which new temperatures are calculated from old temperatures using Eq. (2-17). If the mesh sweep moves from left to right and top to bottom, then any given node will use new temperatures for the neighbors on sides 1 and 4, and old temperatures on sides 2 and 3.

It is desirable to increase the convergence of the iterative procedure described above by "overrelaxing" the temperatures at each iteration. This is done by using the constant, β, in the following manner:

$$T_K^{new} = \beta T_K^n + (1 - \beta) T_K^{n-1} \quad . \tag{2-18}$$

T_K^n are the temperatures calculated during the nth iteration using Eq. (2-17). T_K^{n-1} are the final "overrelaxed, extrapolated" temperatures resulting from the (n-1)th iteration. T_K^{new} are, of course, the "overrelaxed" temperatures being calculated during this nth iteration. The β factor is an input quantity which must lie in the range, $1 \leq \beta < 2$, and usually lies between 1.6 and 1.9 for the best convergence. If a given problem does not converge in one machine run, β may be changed (usually decreased) for the next run.

After the overrelaxation is finished, another acceleration technique, "dominant-error-mode extrapolation," is applied. At each iteration an error term is calculated for the mesh as a whole.

$$e_n = \sum_{K=1}^{K_{MAX}} \left| \frac{T_K^{n-1}}{T_K^n} - 1 \right| . \tag{2-19}$$

Then a ratio of these error terms gives an extrapolation factor, f,

$$f = \frac{e_n}{e_{n-1}} \tag{2-20}$$

which is used to calculate a better estimate of the temperatures.

$$T_K^{final} = T_K^{new} + \frac{f}{1 - f} (T_K^{new} - T_K^{n-1}) . \tag{2-21}$$

T_K^{new} and T_K^{n-1} are defined above in Eq. (2-18). T_K^{final} may now be referred to as the final "overrelaxed, extrapolated" temperatures, T_K^n, for the nth iteration. This extrapolation is not done if the factor, f, in Eq. (2-20) is greater than one. Actually, we found that this extrapolation should only be used every 40th or 50th iteration, depending on the problem, so that the temperatures have a chance to settle down before being kicked again. The frequency of extrapolation is an input quantity which is used for the whole machine run.

Successive iterations are carried out until

$$\max_K \left| \frac{T_K^{n-1}}{T_K^n} - 1 \right| \leq \varepsilon \tag{2-22}$$

where ε is the specified convergence criterion. Note that this error term is calculated for every node in the mesh, but only the maximum is used in Eq. (2-22).

2.4 Numerical Technique for Transient Heat Conduction

The partial differential equation which represents transient heat conduction in an isotropic material may be written in vector notation as

$$\vec{\nabla} \cdot k \vec{\nabla} T + q = \rho C \frac{\partial T}{\partial t} \tag{2-23}$$

where t represents time, ρ is the density, and C is the specific heat (k, T, q, and $\vec{\nabla}$ were defined in Section 2.3). Integrating over the nodal volume, V_K, and using Green's first identity (as was done in Section 2.3) we have

$$\int_{A_K} k \frac{\partial T}{\partial n} dA + \int_{V_K} q \, dV = \int_{V_K} \rho C \frac{\partial T}{\partial t} dV \quad . \tag{2-24}$$

Again we assume that k and $\frac{\partial T}{\partial n}$ do not vary as we integrate over any of the four surface areas (see Fig. 2.6). In addition, we assume that q, ρ, C, and $\frac{\partial T}{\partial t}$ do not vary as we integrate over the nodal volume. The resulting equation is

$$k_K \sum_{S=1}^{4} A_{K,S} \frac{\partial T}{\partial n_{K,S}} + q_K V_K = \rho_K C_K V_K \frac{\partial T}{\partial t} \tag{2-25}$$

where $A_{K,S}$ was defined in Section 2.3.

Next we integrate Eq. (2-25) over the time step, Δt:

$$\int_t^{t+\Delta t} \left(k_K \sum_{S=1}^{4} A_{K,S} \frac{\partial T}{\partial n_{K,S}} \right) dt + \int_t^{t+\Delta t} q_K V_K \, dt = \int_t^{t+\Delta t} \rho_K C_K V_K \frac{\partial T}{\partial t} dt \; . \tag{2-26}$$

Assuming that k, q, ρ, and C do not vary appreciably over Δt, Eq. (2-26) becomes

$$k_K \sum_{S=1}^{4} A_{K,S} \int_t^{t+\Delta t} \frac{\partial T}{\partial n_{K,S}} dt + q_K V_K \Delta t = \rho_K C_K V_K [T_K(t + \Delta t) - T_K(t)] . \tag{2-27}$$

The standard alternating-direction method[1] may now be used by developing two forms of (2-27) to be used alternately in successive time steps. The first finite-difference equation is implicit in one direction (say z) and explicit in the other (say r); in the second equation the directions are interchanged. This procedure results in a set of simultaneous equations which may be solved for new temperatures by the direct inversion of a tridiagonal matrix. No iterations are required. However, when we used this method, we found that the explicit approximations on two of the four sides caused instabilities similar to those in a one-step explicit method. In the conventional ADI technique a term of the form $\left.\frac{\partial T}{\partial n}\right|_{t+\Delta t}$ is neglected on the two explicit faces. By including approximations for these terms, we obtained a set of finite-difference equations that appeared to be fully implicit but which could still be solved by the direct inversion of a tridiagonal matrix.

We assume that $\frac{\partial T}{\partial n}$ varies linearly over a time step and apply the trapezoid rule to Eq. (2-27). Defining $T = T(t)$, $T^{\Delta} = T(t + \Delta t)$, $\frac{\partial T}{\partial n} = \left.\frac{\partial T}{\partial n}\right|_t$ and $\frac{\partial T^{\Delta}}{\partial n} = \left.\frac{\partial T}{\partial n}\right|_{t+\Delta t}$, we then have

$$k_K \sum_{S=1}^{4} A_{K,S} \frac{\Delta t}{2} \left(\frac{\partial T^{\Delta}}{\partial n_{K,S}} + \frac{\partial T}{\partial n_{K,S}} \right) + q_K V_K \Delta t = \rho_K C_K V_K (T_K^{\Delta} - T_K) . \tag{2-28}$$

Putting the $t + \Delta t$ terms on the left we have

$$\rho_K C_K V_K T_K^{\Delta} - \frac{k_K \Delta t}{2} \sum_{S=1}^{4} A_{K,S} \frac{\partial T^{\Delta}}{\partial n_{K,S}} = q_K V_K \Delta t + \rho_K C_K V_K T_K +$$

$$+ \frac{k_K \Delta t}{2} \sum_{S=1}^{4} A_{K,S} \frac{\partial T}{\partial n_{K,S}} . \qquad (2\text{-}29)$$

Using the approximation for $\frac{\partial T}{\partial n}$ given in Eq. (2-15) we have

$$v_K T_K^{\Delta} - \sum_{S=1}^{4} \omega_{K,S} \left(\frac{\tau_{K,S}^{\Delta} - T_K^{\Delta}}{B_{K,S}} \right) = u_K + v_K T_K + \sum_{S=1}^{4} \omega_{K,S} \left(\frac{\tau_{K,S} - T_K}{B_{K,S}} \right) \qquad (2\text{-}30)$$

where u_K, v_K, $\omega_{K,S}$ are defined as

$$u_K = q_K V_K \Delta t \; ; \quad v_K = \rho_K C_K V_K \; ; \quad \omega_{K,S} = \frac{k_K \Delta t A_{K,S}}{2} . \qquad (2\text{-}31)$$

Equation (2-30) is implicit in both the r and z directions so that a complete matrix must be used to solve for all the temperatures simultaneously. However, if we consider only the nodal points in one particular column of the mesh (as shown in Fig. 2.8) and assume that we know the temperatures at time, $t + \Delta t$, on either side of this column, we may then rewrite Eq. (2-30):

$$\left(\frac{-\omega_{K,1}}{B_{K,1}} \right) \tau_{K,1}^{\Delta} + \left(\sum_{S=1}^{4} \frac{\omega_{K,S}}{B_{K,S}} + v_K \right) T_K^{\Delta} + \left(\frac{-\omega_{K,3}}{B_{K,3}} \right) \tau_{K,3}^{\Delta} = u_K + v_K T_K +$$

$$+ \sum_{S=1}^{4} \omega_{K,S} \left(\frac{\tau_{K,S} - T_K}{B_{K,S}} \right) + \sum_{S=2,4} \frac{\omega_{K,S} \tau_{K,S}^{\Delta}}{B_{K,S}} . \qquad (2\text{-}32)$$

This Eq. (2-32) represents a set of IMAX simultaneous equations involving only the temperatures in column J. For an internal point, K, which has nodes above and below it, we can rewrite Eq. (2-32) as

$$a_I \, T^{\Delta}_{K-JMAX} + b_I \, T^{\Delta}_{K} + c_I \, T^{\Delta}_{K+JMAX} = p_I \tag{2-33}$$

or in matrix notation:

$$\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & a_3 & b_3 & c_3 & & \\ & & a_4 & b_4 & c_4 & \\ & & & \cdots & & \\ & & & a_{IMAX} & b_{IMAX} & \end{bmatrix} \begin{bmatrix} T^{\Delta}_{I=1,J} \\ T^{\Delta}_{2,J} \\ T^{\Delta}_{3,J} \\ T^{\Delta}_{4,J} \\ \cdots \\ T^{\Delta}_{IMAX,J} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ \cdots \\ p_{IMAX} \end{bmatrix} \tag{2-34}$$

ORNL-DWG 71-3958

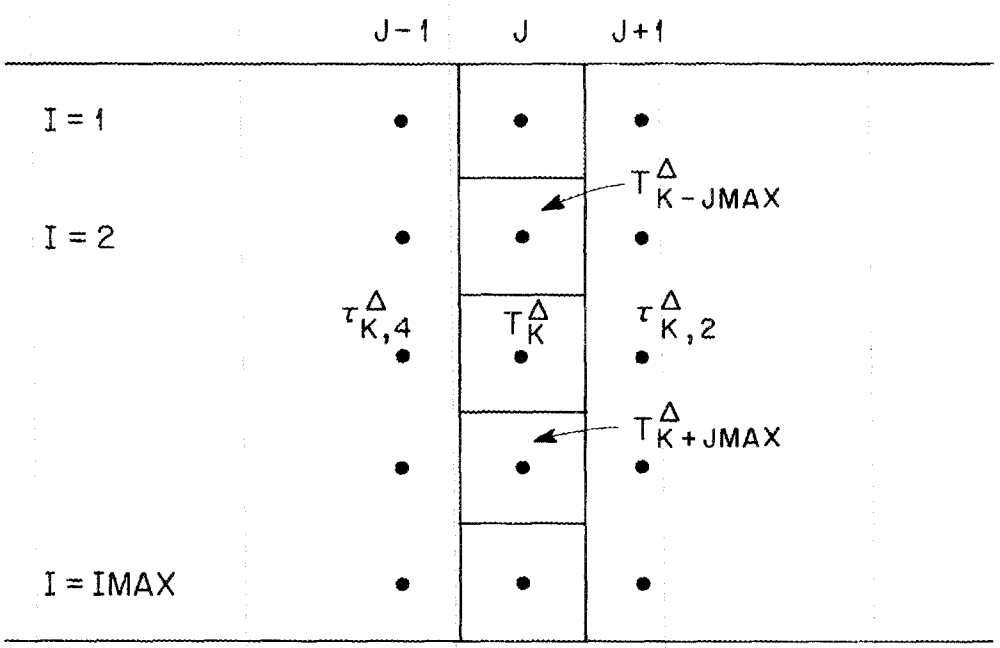


Figure 2.8

The coefficients and $T_{I,J}$ are defined as

$$a_I = \frac{-\omega_{K,1}}{B_{K,1}} \qquad b_I = \sum_{S=1}^{4} \frac{\omega_{K,S}}{B_{K,S}} + v_K \qquad c_I = \frac{-\omega_{K,3}}{B_{K,3}}$$

$$p_I = u_K + v_K T_K + \sum_{S=1}^{4} \omega_{K,S} \left(\frac{\tau_{K,S} - T_K}{B_{K,S}} \right) + \sum_{S=2,4} \frac{\omega_{K,S}\, \tau^{\Delta}_{K,S}}{B_{K,S}} +$$

$$+ \frac{\omega_{K,1}\, \tau^{\Delta}_{K,1}}{B_{K,1}} \text{ (If } I = 1) + \frac{\omega_{K,3}\, \tau^{\Delta}_{K,3}}{B_{K,3}} \text{ (If } I = \text{IMAX)}$$

$$T_{I,J} = T_K \text{ , where } \quad K = (I - 1) * \text{JMAX} + J \quad . \tag{2-35}$$

The only thing remaining to be done is to supply the temperatures, $\tau^{\Delta}_{K,2}$ and $\tau^{\Delta}_{K,4}$, which are used on the right-hand side, p_I, of our matrix equation. Since we move from left to right by columns, the previous column of temperatures, $\tau^{\Delta}_{K,4}$, has already been calculated. Thus we only need an approximation for $\tau^{\Delta}_{K,2}$:

$$\tau^{\Delta}_{K,2} = \frac{\partial \tau_{K,2}}{\partial t} \cdot \Delta t + \tau_{K,2} \quad . \tag{2-36}$$

In evaluating $\frac{\partial \tau}{\partial t}$ let us define $\tau^{-\Delta} = \tau(t - \Delta t)$ so that we have

$$\frac{\partial \tau_{K,2}}{\partial t} = \frac{\tau_{K,2} - \tau^{-\Delta}_{K,2}}{\Delta t} \quad . \tag{2-37}$$

Substituting Eq. (2-37) back into Eq. (2-36), we now have an approximation for $\tau^{\Delta}_{K,2}$ which can be used in calculating p_I in Eq. (2-35). When using this approximation for $\tau^{\Delta}_{K,2}$ to solve real heat problems, we found that the results were not quite good enough (1-2% error). Therefore it was necessary to repeat our mesh sweep a second time using the temperature just calculated for $\tau^{\Delta}_{K,2}$ rather than the approximation in Eq. (2-36).

The equations for the row sweep are the same as those for the column sweep, except for the interchange of references to faces 2 and 3 (right and lower sides) and faces 4 and 1 (left and upper sides). There are JMAX equations, one for each row.

To summarize the transient calculations, we first move across the mesh, column by column, solving the tridiagonal system for each column of temperatures. It is necessary to repeat this column-by-column sweep a second time at the same time step. We then increment the time by Δt and move down the mesh row by row solving for each row of temperatures. Again, it is necessary to repeat the row sweep a second time at the same time step. The time is again incremented by Δt and we start all over with the column sweep, and so on, until the desired time period has been spanned.

2.5 Position, Time, and Temperature Dependent Functions

There are four quantities which may be functions of position, time, or temperature in both the steady-state and the transient programs. These four quantities are the thermal conductivity, k, the specific heat, C, the density, ρ, and the heat-generation rate, q. In addition, the transient code allows the boundary conditions to vary at any time step. When any one of these quantities, such as thermal conductivity, is to be treated as an external function, the FORTRAN program calls a user-written subprogram to calculate the appropriate conductivity value. This FUNCTION subprogram is called at every iteration for each node in the mesh. Actually, a skeleton subprogram exists for each of the quantities, and the user needs only to add the necessary FORTRAN statements describing his particular function. Upon entry to any of these subprograms, all variables within the program (such as nodal position, nodal material, elapsed time, nodal temperature, time increment, etc.) are available to the user. Thus the thermal conductivity of a given node may be a function of that node's temperature or position, or it may vary with time. Rather than using some type of equation to calculate the thermal conductivity, the user may wish to input a table of thermal conductivities versus temperature so that he can calculate the proper conductivity using linear interpolation. The user indicates in the input data those quantities which are to be external functions, and the rest remain constant throughout the problem.

When using temperature-dependent properties in the transient code it is necessary to approximate the temperature, $T(t + \Delta t)$, using the following equation:

$$T(t + \Delta t) = T(t) + \frac{\partial T}{\partial t} \cdot \Delta t \qquad (2\text{-}38)$$

where $\frac{\partial T}{\partial t}$ is calculated at time, t. This approximation has already been coded in the skeleton subprogram so that the user needs only to evaluate his function using the $T(t + \Delta t)$ supplied. In steady-state problems the user should evaluate his temperature-dependent function using the most recent value of the temperature, T^{new} (cf Eq. (2-18)).

When the boundary conditions are allowed to vary in a transient problem, the user must specify two things. First, he must specify the time steps at which the boundary conditions are to change, and second, he must specify a numerical quantity to be used in calculating this change. This numerical quantity may be used within the skeleton subprogram in any way the user wishes. At present, the subprogram is set up to treat this numerical quantity as either a new boundary value replacing the old one, or as a rate of change in the current boundary values. There are two types of boundary values which may be changed within this subprogram. The first is an outside temperature, either the external temperature, T_{ext}, or the surface temperature, T_b. The second is the effective boundary layer thickness, 1/a, explained in Section 2.1. There is one subprogram which is called when changing the outside temperature and another subprogram called when changing the boundary layer thickness. Both subprograms are called only once at the beginning of each time step so that the boundary conditions may be changed for any of the nodes along the outside edges. By letting the boundary layer thickness (effectively, h) be temperature dependent it is very easy to consider a radiation-type boundary condition. In terms of h we simply let

$$h = \sigma F_{se}(T_b^4 - T_{ext}^4) / (T_b - T_{ext}) = \sigma F_{se}(T_b^2 + T_{ext}^2)(T_b + T_{ext}) \quad (2\text{-}39)$$

where σ = Stefan-Boltzmann constant, F_{se} = radiation interchange factor, T_b = temperature of boundary surface, and T_{ext} = external temperature.

There is a total of six different subprograms which may be supplied by the user to implement the various position-, time-, and temperature-dependent functions previously described. These six subprograms are listed in Table 2.1 according to the physical quantity which each is supposed to calculate. A skeleton subprogram has been written for each of these six subprograms so that the user will already have the calling sequence and preliminary calculations set up. A FORTRAN listing of each subprogram is given in Appendix E. The commentary in these listings should prove helpful to the user. There are two listings for the CONDF function, one for the steady-state and another for the transient.

Table 2.1

Physical Quantity To Be Calculated		Subprogram Called
1. Thermal Conductivity	- k	CONDF
2. Specific Heat	- C	SHEATF (Transient only)
3. Density	- ρ	SMASSF (Transient only)
4. Heat-Generation Rate	- q	QF
5. Boundary Temperature	- T_b or T_{ext}	TOBARF (Transient only)
6. Boundary Layer Thickness	- 1/a	ARATEF (Transient only)

When solving heat-conduction problems involving temperature-dependent properties, it should be pointed out that the numerical procedures described in Sections 2.3 and 2.4 are inaccurate when severe thermal gradients

arise.[6] Equations (2-13) and (2-25) assume that the thermal conductivity, k_K, of node K is the same on all four faces of the node. This assumption is not correct when the temperatures on the four faces of the node are quite different. Therefore we have included a procedure in the steady-state code to calculate the conductivity on the node interfaces as a function of the interface temperatures. Of course this procedure is only used when the conductivity is treated as an external function (CONDF is used). The interface temperatures are calculated by Eq. (2-40) below.

$$T_{face} = T_K + \frac{\partial T}{\partial x} \cdot \Delta x \quad . \tag{2-40}$$

We plan to include this procedure as an option in the transient code so that the effects of variable conductivity in problems with steep temperature gradients can be investigated. One-dimensional calculations[6] have demonstrated that serious errors can occur.

2.6 Approximation of Nodal Melting and Freezing

In many transient problems it is desirable to consider the effect of nodal melting and freezing within certain materials. We have included an approximate treatment of the change of state. The user must first specify which materials may melt, along with their melting temperature, T_m, and latent heat of fusion, H_{fu}. Then as the temperature within a meltable node reaches its melting point, a heat bank, H, is set up for that node to absorb the latent heat of fusion as it changes to a liquid. While the node changes state its temperature remains constant at the melting temperature given. The equation describing the amount of energy absorbed by the heat bank at each time step is

$$\Delta H = C \cdot (T_{proj} - T_m) \tag{2-41}$$

where T_{proj} is the projected temperature which would be reached within node, K, if no heat were lost to the heat bank. Eventually the amount of energy absorbed by the heat bank equals or exceeds the latent heat of fusion for that particular material. At this point the temperature of node, K, is set to:

$$T_K = T_m + (H - H_{fu})/C \tag{2-42}$$

and new properties (k, ρ, C) characteristic of the liquid material are assigned to that node. The temperature is now allowed to rise, as before, according to the laws of conduction. During the time of melting or freezing it might be helpful to use shorter time steps, thus improving the approximations involved.

Of course, a liquid material will transfer heat by means of natural convection as well as conduction. The hotter liquid will move upward while the cooler liquid will fall due to their difference in densities. Thus the user may wish to account for this extra transfer of heat by using a thermal conductivity which is slightly higher than that of the liquid material. The best way to simulate natural convection in this program would be to let the liquid conductivity be a function of temperature differences between nodes. Then the higher temperature gradients would create greater conductivities (and vice versa) with the normal liquid conductivity as a floor.

The process of freezing is handled in exactly the same way as melting except that energy is removed from the heat bank rather than added. This situation will arise when a material first becomes hot enough to melt and then later cools off enough to freeze, or is initially molten and freezes as the assembly cools.

3. INPUT DESCRIPTION

3.1 General

All of the input for both the transient and steady-state codes has been designed in an easily usable free-form style with keywords identifying each item of information. Since there are several different types or categories of information which must be supplied, we have divided the input into seven sections (only six sections are used in steady-state problems). To show the type of information which must be supplied in each section, let us consider Fig. 3.1 below.

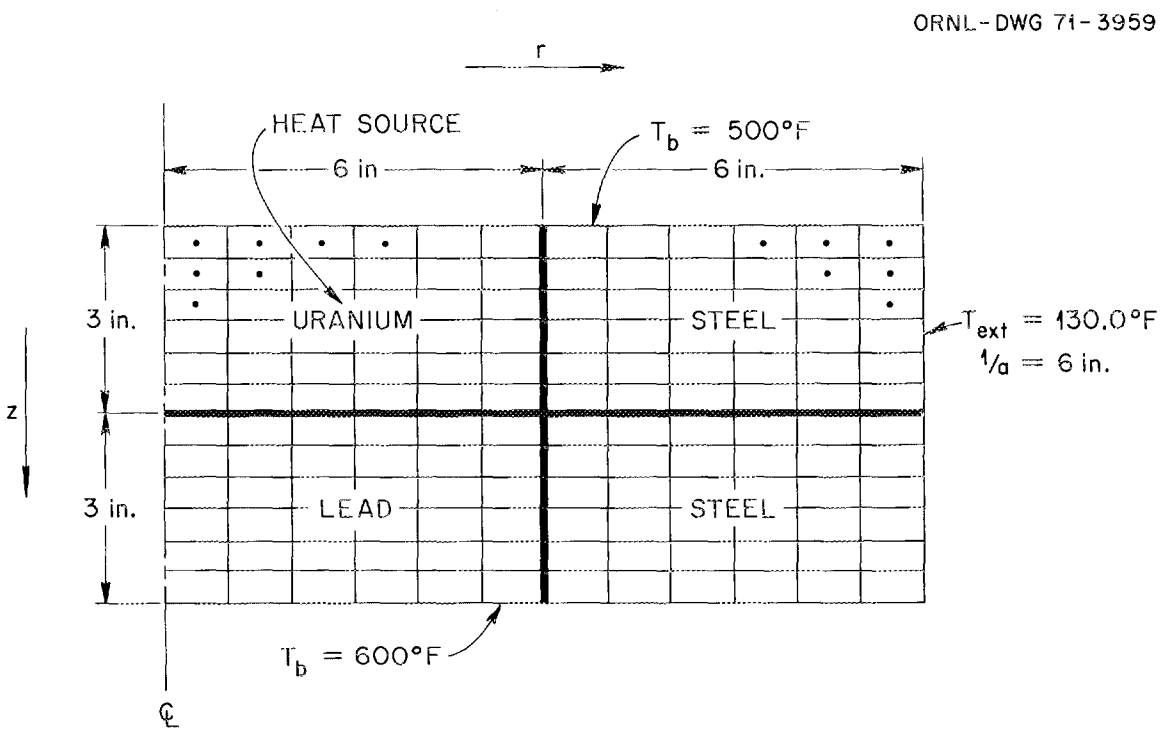


Figure 3.1

This particular example solves the transient problem for the hypothetical cylinder shown in Fig. 3.1. The initial temperatures are specified as 130.0°F for every node in the mesh, and the problem is to be run for five minutes with temperatures calculated every second (300 time steps). The input which would be required to run this problem is given in Fig. 3.2.

Each section contains information dealing with a different aspect of the problem. The alphabetic keywords are used to identify the information given and where it is to go within the mesh. The numeric values may be followed by a units field such as B/P'F, which represents BTU per pound per degree Fahrenheit. All of the input given in Fig. 3.2 could be punched on cards as shown, and it would be acceptable to the transient code. Each of the seven input sections must begin with a section definition card (***Section Name***), and each ends when the next section definition card is read. These sections may appear in any order except for the first two. Within any of these seven sections (except the ***MESH*** section), the various items, themselves, may appear in any order. Thus in Fig. 3.2 we could have given the HEAT SOURCE value before the MATERIAL NO specifications. Each section has a certain syntactical format which uses keyword names, equal signs, and numeric values. These syntactical formats are called expressions, and there are three different types of these expressions. The first and simplest is the "arithmetic-expression" of the form:

$$\begin{array}{ccc} \underline{\text{exp-name}} & = & \underline{\text{values}} \\ \text{HORZREGS} & = & 2 \end{array} \qquad (3\text{-}1)$$

where values are either numeric or alphabetic (YES, NO, ...). The second type of expression is the "name-expression" which is of the form:

```
Title Card :  TRANSIENT PROBLEM TITLE CARD - 4 REGIONS, 144 NODES

Section 1  :                    ***HEADER***

              INITIALZ = YES   LISTINPT = YES  LISTMESH = YES  LISTTEMP = YES
              LISTINCR =   1   SAVETEMP =  NO  SAVEUNIT  =  0  LENGTH  = FEET
              MASS  =  POUND   TIME  =   HOUR  TEMP = 'FAHREN  HEAT    =  BTU
              STARTIME = 0 H.  STRTSTEP  =  1  NTIMSTEP = 300  MAXSEC   =   0

Section 2  :                     ***MESH***

              SYSTEM = R,Z        HORZREGS = 2       VERTREGS = 2
              HORZDIST = 6.0 I.       6.0 I.
              HORZNODE =   6            6
              VERTDIST = 3.0 I.       3.0 I.
              VERTNODE =   6            6

Section 3  :                     ***NAMES***

              MATERIAL NO = 1  FOR  URANIUM ,
                            2  FOR  LEAD    ,
                            3  FOR  STEEL   .

Section 4  :                   ***BOUNDARY***

              EDGE TEMP = 500.0 'F.  FOR  TOP.SIDE ,
                          600.0 'F.  FOR  BOT.SIDE .
              EXT  TEMP = 130.0 'F.  FOR  RHS.SIDE .
              GRAD DIST =   6.0  I.  FOR  RHS.SIDE .

Section 5  :                  ***MATERIALS***

              MATERIAL NO  =    1          FOR  R,Z.REGION (1,1) ,
                                2          FOR  R,Z.REGION (2,1) ,
                                3          FOR  COL.REGION  2    .
              CONDUCTIVITY = 10.0 B/FH'F.  FOR  MAT.NOS     1    ,
                             15.0 B/FH'F.  FOR  MAT.NOS     2    ,
                             20.0 B/FH'F.  FOR  MAT.NOS     3    .
              DENSITY      =  100 P/FFF.   FOR  MAT.NOS     1    ,
                              200 P/FFF.   FOR  MAT.NOS     2    ,
                              300 P/FFF.   FOR  MAT.NOS     3    .
              SPHEAT       =   .1 B/P'F.   FOR  MAT.NOS     1    ,
                               .2 B/P'F.   FOR  MAT.NOS     2    ,
                               .3 B/P'F.   FOR  MAT.NOS     3    .
              HEAT SOURCE  =   20 B/HFTF.  FOR  R,Z.REGION (1,1) .

Section 6  :              ***TEMPERATURES***

              TEMPERATURES     =  130.0 'F.  FOR  ALL.MESH .
              MAX TEMP ALLOWED =  800.0 'F.  FOR  ALL.MESH .

Section 7  :                     ***TIME***

              TIME STEP INCR = 1.0 S.  FOR TIM.STEPS 1-300 .
              TEMP LIST FREQ =  60     FOR ALL.TIME .

Endata Card:                   ***ENDATA***
```

Figure 3.2

The second type of expression is the "name-expression" which is of the form:

```
exp-name    =  value  FOR  material-name  .
-----------    -----  ---  -------------
MATERIAL NO =    1    FOR    URANIUM     .
```
(3-2)

where material-name is an eight- (or less) character alphanumeric name. The third and most complicated type of expression is the "for-expression" which is of the form:

```
exp-name     =   value      FOR  location-name  integer-exp  .
-----------     ----------  ---  -------------  -----------
SPHEAT       =  .1 B/P'F.   FOR   MAT.NOS           1        .
HEAT SOURCE  =  20 B/HFFF.  FOR   R,Z.REGION       (1,1)     .
TIME STEP    =   1 S.       FOR   TIM.STEPS         1-300    .
```
(3-3)

A more complete definition and description of these three expressions is given in Section 3.2.

Although most users will probably include all seven sections in their input, it is possible to omit any of the sections or any of the items within a section when desired. When this is done, default values will be used for those variables which are undefined. Table 3.1 lists all of the input sections, the type of expression used in each, and their respective sequence in the input stream.

Table 3.1

No.	Section Name	Expression-Type	Required	Sequence
First.	Title Cards		Yes	First
1.	***HEADER***	Arithmetic-Exp	Optional	Second
2.	***MESH***	Arithmetic-Exp	Optional	Third
3.	***NAMES***	Name-Exp	Optional	Any Order
4.	***BOUNDARY***	For-Exp	Optional	Any Order
5.	***MATERIALS***	For-Exp	Optional	Any Order
6.	***TEMPERATURES***	For-Exp	Optional	Any Order
7.	***TIME***	For-Exp	Optional	Any Order
Last.	***ENDATA***		Yes	Last

In describing the format of each of these expression-types, it should be stressed that everything is free-form. This means that the various input entities (names, numbers, equal signs, periods, etc.) may go anywhere on the card as long as they are separated from their neighbors by one or more blanks. Thus the blank which is between the keyword, INITIALZ, and its equal sign in Fig. 3.2 is absolutely necessary.

A detailed description of each of these seven sections is given in Sections 3.4-3-10. Any differences between steady-state input and transient input will be explained in the appropriate section. Section 3.3 explains the use and choice of available units. Many times a user wishes to run the same problem several times with slight variations in some of the parameters. This is discussed in Section 3.11. Section 3.12 explains how to start (or restart) a problem from temperature distributions previously saved on cards, tape, or disk. The maximum problem sizes which are currently allowed are given in Section 3.13. These limits may be increased if necessary.

3.2 Expressions

As was explained previously, all input to these two codes must be written in the form of alphanumeric expressions. There are three types of expressions which are used during the preparation of the seven input sections. However, only one type of expression is allowed in each input section as shown in Table 3.1. Before explaining each expression-type individually, let us define the notation which will be used. Whenever items appear within brackets [ ] they are optional and may be omitted at the user's discretion. Whenever items appear within braces { } they are required and must not be omitted. If items are stacked within a pair of brackets or braces $\left\{\begin{matrix}\text{item 1}\\ \text{item 2}\\ \text{item 3}\end{matrix}\right\}$, this means that one item may be chosen from the stack. If an item is not enclosed within any pair of brackets or braces, this means that it is ignored (treated as commentary) while the expression is being deciphered. Again it should be stressed that every item must be separated from its neighbor by one or more blanks. Also numeric values, unless otherwise specified, may be written as fixed or floating-point numbers with or without an exponent (E or D).

Using the notation defined above, we may write the general form of the "arithmetic-expression" as:

	{exp-name}	{=}	{value}	[unit]	{value}	[unit]	...	
example 1:	HORZDIST	=	6.0	I.	6.0	I.		(3-4)
example 2:	SAVETEMP	=	NO					

The expression-name (a 1-8 character keyword) always comes first followed by an equal sign and then the proper number of alphabetic or numeric values. Certain types of numeric values may have units attached to specify length,

time, etc. When alphabetic values follow the equal sign, they must conform to what is expected by the particular input section in which they are used. Note that the expression-name must be spelled exactly right, and there is no final period at the end of the expression.

The next type of expression which is slightly more complicated is the "name-expression" written in general form as:

	[exp-name]	=	{integer value}	{FOR}	{material-name}	{',}	
example 1:	MATERIAL NO	=	1	FOR	URANIUM	,	
			2	FOR	LEAD	.	(3-5)
example 2:	MATERIAL NO	=	3	FOR	STEEL	.	

Here the expression-name is required only at the beginning of the expression and need not appear again as long as commas are used as separators (see example 1 above). This allows the user to specify several material-names and numbers in one expression. Another important difference between the expression-name used here and that described earlier is that only the first three characters (MAT) are needed here. Everything after these three characters is ignored until the numeric value is read. Thus the equal sign is not really needed except for readability. The value itself should always be a positive integer without any units field attached. Following the numeric value should eventually come the word FOR. Immediately following the FOR should come a 1-8 character material-name. This material-name may contain any number of alphanumeric characters except blanks. Only the first eight characters are retained. Following the material-name should come a comma (indicating continuation of the expression) or a period (indicating the end of the expression). If the expression is continued, as indicated by a comma, the next item expected is

another numeric value. At present the "name-expression" is used only in the ***NAMES*** section of the input data.

The last and most complicated type of expression is the "for-expression" which may be written in general form as:

[exp-name]	=	{value}	[units]	FOR	{loc-name}	[integer-exp]	{'}
example 1: TEMPS	=	130	'F.	FOR	ALL.MESH		.
example 2: CONDUCTIVITY	=	10.0		FOR	MAT.NOS	1	.
example 3: TIME STEP INCR	=	1	S.	FOR	TIM.STEPS	1-300	.
example 4: HEAT SOURCE	=	2.0E 01	B/HFFF.	FOR	R,Z.REGION	(1,1)	.
example 5: EDGE TEMP	=	500.0	'F.	FOR	TOP.SIDE		,
		600.0	'F.	FOR	BOT.SIDE		.

(3-6)

The expression-name is treated the same way here as it was in the "name-expression." Only the first three letters are significant, and the name may be omitted when the expression is continued with a comma. The value in this case may be any legal fixed or floating-point number as shown in the examples above. The units field following the numeric value is optional and need only be used when the input units differ from the standard internal units set in the ***HEADER*** section. Notice that the last character of every units field must be a period. The next item of information looked at after the units field is the location-name. The location-name is always a seven-character name with a period in the fourth position. Extraneous characters on the end of the name are ignored (ALL.MESH). Each input section has its own set of unique location-names which are acceptable only to it. As additions and improvements are made to these two heat transfer codes, new names will be added to the appropriate sets to allow for more flexible input.

Integer-expressions may then follow the location-name to specify in what particular location the numeric value is to be placed. These integer-expressions, when used, come in two types: those without parentheses (type-1) and those within parentheses (type-2). The various forms of each type are shown below.

	Type-1 Integer-expressions	Type-2 Integer-expressions	
	i i i ...	(i,i) ...	
		(i,i-i) ...	
	i-i i-i i-i ...	(i-i,i) ...	
		(i-i,i-i) ...	(3-7)
example 1:	1 2 3 8 9 10 13 14	(2,3) (2,4) (3,3) (3,4)	
example 2:	1 2 3 8 9 10 13-14	(2,3-4) (3,3-4)	
example 3:	1 2 3 8-10 13-14	(2-3,3) (2-3,4)	
example 4:	1-3 8-10 13-14	(2-3,3-4)	

where i represents any positive integer. Type-1 integer-expressions consist of simple integers written one after another. Two integers may be separated by a hyphen (8-10) to indicate the lower and upper limit of a range of integers (8 9 10). The type-2 integer-expressions are more complicated. The first integer (or range of integers) within the parentheses represents a row number in the mesh, and the second integer (or range of integers) represents a column number in the mesh. Thus the type-2 integers are used only to specify locations within the mesh while type-1 integers may be used to specify many different things (time steps, material numbers, etc.). Notice in the two sets of examples given above that the same set of integers has been represented in four different ways. Following the last integer-expression should come a comma (indicating continuation of the "for-expression") or a period (indicating the end of the "for-expression").

These three types of expressions have been designed to make the preparation of input as easy as possible for the user. Rather than forcing the user to remember what type of input data goes in which columns on what cards, we have allowed the use of keywords to indicate what type of data is being used. The data itself may be placed anywhere on the card, and the cards may appear in any order (with respect to keywords) in the input stream.

3.3 Units

Many of the numeric quantities which must be supplied as input to these two codes have dimensions or units associated with them to specify length, time, etc. To facilitate the use of such units during the preparation of input, we have allowed a units field to be attached to any numerical value. The user simply follows the numerical value with its appropriate units field, leaving one or more blanks between the two. As has been stated previously, the last character of the units field must be a period. The field itself is made up of alphabetic characters plus two special characters: a single quote mark, ', and a divide sign,/. Each alphabetic character represents one of the nineteen different units which are acceptable to these two codes. These nineteen units fall into five general categories or classes: length, mass, time, temperature, and heat. Table 3.2 gives a breakdown of the units and their corresponding characters within each of the five classes.

Table 3.2

Length	Char	Mass	Char	Time	Char
Inches	I	Pound	P	Second	S
Feet	F	Gram	G	Minute	N
Centimeter	C	Kilogram	K	Hour	H
Meter	M				

Temp	Char	Heat	Char
Fahrenheit	'F	BTU	B
Rankine	'R	Calorie	Q
Centigrade	'C	Joule	J
Kelvin	'K	Erg	E
		Watt	W

Notice that the first letter of each word has been used as the alphabetic character to represent that unit (with two exceptions: minutes, N, and calories, Q). Under the temperature class each of the characters has been preceded by a single quote mark. This quote mark stands for the degree sign, °, and must be used whenever temperature units are being specified. The other special character is the divide sign, /, which is used to indicate that certain units belong in the denominator of the units field. For example, the divide sign in the units field, B/FH'F., indicates that feet, hours, and degrees Fahrenheit belong in the denominator while BTU belongs in the numerator. In Table 3.3 various physical quantities are shown with their corresponding unit fields given in both the English and metric systems.

Table 3.3

Physical Quantities	English (In. Sec)	English (Ft. Hr)	Metric (CGS)	Metric (MKS)
Distance	I.	F.	C.	M.
Conductivity	B/IS'F.	B/FH'F.	W/C'C.	W/M'C.
Heat Source	B/IIIS.	B/FFFH.	W/CCC.	W/MMM.
Temperature	'F.	'F.	'C.	'C.
Density	P/III.	P/FFF.	G/CCC.	K/MMM.

Notice that to specify cubic feet, we must write FFF rather than ft^3, as is normally done. Also the user is not restricted to using only English or only metric units as was done in Table 3.3 above. He may use any combination of units desired. The programs will read the input units given and convert them to the standard internal units set in the ***HEADER***

section. The way in which these internal units are set is shown in Fig. 3.2 under the ***HEADER*** section. If the user asks that his input data be printed out, he will receive all numerical values twice, first using his own input units and second using the converted internal units. All later printout, including temperature distributions, will then use the standard internal units.

In Table 3.2, one of the choices for the unit of heat is watts, W. Actually watts are a measure of power and represent heat per unit time (1 watt = 1 joule/sec) rather than just heat. Therefore whenever watts are involved in a conversion from one set of units to another, it is necessary to consider both heat and time conversions. In the printout however only the heat conversion will be indicated although the time conversion will also be done. For example, in converting heat source values from metric to English units, the printout would indicate that W/MMM was converted to B/FFF. The actual conversion would be from W/MMM to B/HFFF.

Again it should be stressed that a final period is required as the last character in any units field. The maximum number of characters allowed in any one units field is eight. (The longest one, which is for thermal conductivity, requires only seven characters.) If numerical quantities are input without units, then no conversion is done and they are assumed to already be in the same units as the standard internal units.

3.4 ***HEADER*** Section

The first section in the input stream should be the ***HEADER*** section. This section immediately follows the title cards (up to 12 title cards are allowed) as shown in Table 3.1. The beginning of this input section, and any other section, is indicated by a section definition card (***HEADER***). Only the first three asterisks and the first three letters are looked at to determine the name of the section. Actually the section definition name is a free-form field just like everything else, and it may go anywhere on the card. The input data may even follow the section name on the same card as long as blanks separate the fields. The information specified in the ***HEADER*** section usually consists of problem parameters and indicators such as maximum execution time, internal units, printout types, etc. This type of information must be known before the other input sections can be read in.

The only type of expression allowed in the ***HEADER*** section is the "arithmetic-expression." There are twenty-six different keywords which may be used as expression-names in these arithmetic-expressions. Twenty-three of them are acceptable to the steady-state code and seventeen are acceptable to the transient code (fourteen are acceptable to both). Using the "stacked-items" notation defined earlier, the general form of these twenty-six quantities may be written as shown in Table 3.4. In this table "i" represents a positive integer and "f" represents any floating-point number. (Floating-point numbers may also be written as integers.) Notice that in every group of stacked items one item is underlined. This underlined item is the default value assigned to that particular parameter when the user does not specify an overriding value. Many of the values

Table 3.4

HEADER

INITIALZ	= {YES / NO}	LISTINPT	= {YES / NO}	LISTMESH	= {YES / NO}
LISTTEMP	= {YES / NO}	LISTINCR	= {i / 1}	MAXITR	= {i / 500}
MAXSEC	= {i / 0}	SAVETEMP	= {YES / NO}	SAVEUNIT	= {i / 0}
LENGTH	= {I / F / C / M}	MASS	= {P / G / K}	TIME	= {S / N / H}
TEMP	= {'F / 'R / 'C / 'K}	HEAT	= {B / Q / J / E / W}		

Transient Only:

STARTIME	= {f [units] / 0.0}	STRTSTEP	= {i / 1}	NTIMSTEP	= {i / 1}

Steady State Only:

BETA	= {f / 1.5}	EPSILON	= {f / .00001}	EXTRINCR	= {i / 0}
HBALINCR	= {i / 0}	TRAVPLOT	= {CRT / MECH / NO}	MAPPLOT	= {CRT / MECH / NO}
NTRAVSES	= {i / 0}	PLOTCOLS or PLOTROWS	= {i i i ... / 0 0 0 ...}		

on the right-hand side of the equal sign are alphabetic words. These words must be spelled exactly as shown in the table with one general exception. Any alphabetic value may have extra letters added onto the end of it as long as the preceding letters are spelled correctly. For example, in specifying the units of mass the word POUNDS could be used as long as the letter P were the first letter. The extra letters on the end are ignored.

Many of the parameters in Table 3.4 are self-explanatory especially if previous parts of this report have already been read. Thus some of these twenty-six parameters will only be explained briefly, and the user will be referred to other parts of the report. The explanation of those parameters controlling printout is more easily understood by looking at the sample printout in Section 4. The following list discusses each of the twenty-six parameters.

1. INITIALZ: Indicates whether or not to initialize all variables before beginning a particular problem. It is normally YES unless several problems are run together. (See Section 3.11.)

2. LISTINPT: Indicates whether or not to list the input data as it is read in.

3. LISTMESH: Indicates whether or not to print a map of the total mesh indicating nodal and region boundaries, location of all materials, and location of internal heat sources. A table of the material properties is also printed out if it is YES.

4. LISTTEMP: Indicates whether or not to print, at specified time steps in the transient code, a full temperature distribution or just a temperature map. (The chosen time steps are specified in the ***TIME*** input section.) In the steady-state code it indicates whether or not to print a full temperature distribution at specified iterations along the way to convergence. (LISTINCR indicates these chosen iterations.) In the transient code it is normally YES, and in the steady-state code it is normally NO.

5. LISTINCR: Indicates how frequently a single line of temperature information is to be printed out. For example, if it were 5 then a single line of information would be printed every 5th time step (every 5th iteration in steady state). In steady state a full temperature distribution will also be printed with each line if LISTTEMP = YES. (The printout of temperature distributions in the transient code is controlled by input data in the ***TIME*** section.)

6. MAXITR: The maximum number of time steps (in transient) or mesh iterations (in steady state) allowed.

7. MAXSEC: The maximum computer time (in seconds) allowed during execution. If it is zero then no check is made to see if the problem is running longer than expected.

8. SAVETEMP: Indicates whether or not to save all the temperatures on some output device when the problem is finished or when MAXITR or MAXSEC has been exceeded. (See Section 3.12.)

9. SAVEUNIT: The logical output device on which temperatures are saved when SAVETEMP = YES. (See Section 3.12.)

10. LENGTH: The standard internal unit to be used for length. (See Section 3.3.)

11. MASS: The standard internal unit for mass. (See Section 3.3.)

12. TIME: The standard internal unit for time. (See Section 3.3.)

13. TEMP: The standard internal unit for temperature. (See Section 3.3.)

14. HEAT: The standard internal unit for heat. (See Section 3.3.)

15. STARTIME: The initial starting time (t_o) for a transient problem. It is normally zero unless a previous run is being continued where it left off. (See Section 3.12.)

16. STRTSTEP: The initial time step number to be used in starting a transient problem. It is normally one (the first time step) unless a previous run is being continued where it left off. (See Section 3.12.)

17. NTIMSTEP: The total number of time steps to be used in a transient problem.

18. BETA: The relaxation factor ($1 \leq \beta < 2$) used to accelerate steady-state convergence. (See Eq. 2-18.)

19. EPSILON: The tolerance value (maximum acceptable error) tested against in achieving steady-state convergence. (See Eq. 2-22.)

20. EXTRINCR: The frequency at which extrapolation is to occur (every EXTRINCR iteration) in the steady-state code. If it is zero then no extrapolation is done. (See Eq. 2-21.)

21. HBALINCR: The frequency at which regional heat imbalances are to be calculated (every HBALINCR iterations) in the steady-state code. The three worst regions will have their heat imbalances printed out. In the future we may be able to drive these imbalances to zero by adjusting the temperatures within each region accordingly. If HBALINCR = 0 then no heat imbalances are calculated.

22. TRAVPLOT: Indicates whether a traverse-type plot is desired when the steady-state problem is finished. In addition it indicates whether to use the cathode ray tube plotter or the mechanical plotter. Up to five different columns or rows of nodal temperatures may be plotted on one graph. (See Fig. 4.10.)

23. MAPPLOT: Indicates whether a map-type plot is desired when the steady-state problem is finished. In addition it indicates whether to use the cathode ray tube plotter or the mechanical plotter. The complete temperature distribution is plotted in a maplike framework with individual letters representing temperature ranges. (See Fig. 4.9.)

24. NTRAVSES: The number of columns (or rows) which are to be plotted on one traverse-type plot. The maximum number of traverses allowed in any one problem is five. The plot may consist of either columns or rows, but not both.

25. PLOTCOLS or : 26. PLOTROWS The actual column (row) numbers to be plotted on the traverse-type plot. Up to five different column (row) numbers may be specified. Figure 2.3 shows the J-column (I-row) numbers on the mesh.

3.5 ***MESH*** Section

The second section in the input stream should be the ***MESH*** section. This section immediately follows the ***HEADER*** section as shown in Table 3.1. The section definition card which must come before the actual mesh data is ***MESH***. The information specified in this section describes the overall mesh layout. The geometry type, the number and size of the regions, and the number of nodes within each region are given here. Figures 2.2 and 2.3 show the various distances (h,H) and numbers (m,n) which must be specified. In addition to describing the mesh layout, this section may also be used to specify the direction of the steady-state mesh sweeps. A user may alternate the direction of successive mesh sweeps if desired. For example, he may wish to first sweep through the mesh by rows starting at the upper left corner, and then sweep through the mesh by columns starting at the upper left corner, and then sweep through the mesh by columns starting at the lower right corner. This is done by using the MSWEEP parameters shown in Table 3.5.

As in the ***HEADER*** section, only "arithmetic-expressions" are allowed here. There are nine different parameters which may be specified in this section. Seven of them are acceptable to the transient code while all nine of them are valid in the steady-state code. Table 3.5 lists all nine parameters in the "stacked-item" notation described earlier. Note that the right-hand side values, LEF and RIG, could actually be written as LEFT and RIGHT, since only the first three characters are looked at. As before, "i" represents a positive integer and "f" represents any floating-point number. The underlined items are the default values assigned when none are given by the user. In all input sections

except this one the various parameters may appear in any order. However, in this one section the HORZREGS and VERTREGS must come before the HORZDIST, VERTDIST, HORZNODE, and VERTNODE parameters. This is because the number of regions must be known before the proper number of distances and node numbers can be accepted.

Table 3.5

MESH

$$\text{SYSTEM} = \left\{\begin{matrix}\underline{R,Z}\\ R,T\\ Y,Z\end{matrix}\right\} \qquad \text{HORZREGS} = \left\{\begin{matrix}i\\ \underline{0}\end{matrix}\right\} \qquad \text{VERTREGS} = \left\{\begin{matrix}i\\ \underline{0}\end{matrix}\right\}$$

$$\text{HORZDIST} = \left\{\begin{matrix}f\ [\text{units}] & f\ [\text{units}] & f\ [\text{units}] & \cdots\\ \underline{0} & \underline{0} & \underline{0} & \cdots\end{matrix}\right\}$$

$$\text{VERTDIST} = \left\{\begin{matrix}f\ [\text{units}] & f\ [\text{units}] & f\ [\text{units}] & \cdots\\ \underline{0} & \underline{0} & \underline{0} & \cdots\end{matrix}\right\}$$

$$\text{HORZNODE} = \left\{\begin{matrix}i & i & i & i & \cdots\\ \underline{0} & \underline{0} & \underline{0} & \underline{0} & \cdots\end{matrix}\right\}$$

$$\text{VERTNODE} = \left\{\begin{matrix}i & i & i & i & \cdots\\ \underline{0} & \underline{0} & \underline{0} & \underline{0} & \end{matrix}\right\}$$

<u>Steady State Only</u>:

$$\text{MSWEEP1} = \left\{\begin{matrix}\underline{TOP}\\ BOT\end{matrix}\right\} \quad \left\{\begin{matrix}\underline{LEF}\\ RIG\end{matrix}\right\} \quad \left\{\begin{matrix}\underline{ROW}\\ COL\end{matrix}\right\}$$

$$\text{MSWEEP2} = \left\{\begin{matrix}\underline{TOP}\\ BOT\end{matrix}\right\} \quad \left\{\begin{matrix}\underline{LEF}\\ RIG\end{matrix}\right\} \quad \left\{\begin{matrix}\underline{ROW}\\ COL\end{matrix}\right\}$$

The following list discusses each of the nine parameters in detail. It would be very helpful to refer to Figs. 2.2 and 2.3 while studying this list since they show the actual distances and number of nodes involved.

1. SYSTEM: Indicates the type of geometrical system to be used. R,T stands for the r, θ type geometry. In the Y,Z system (same as x-y), the Y axis replaces the R axis and the Z axis stays the same.

2. HORZREGS: The number of horizontal regions, LHMAX, across the mesh.

3. VERTREGS: The number of vertical regions, LVMAX, down the mesh.

4. HORZDIST: The thickness, H, of each horizontal region across the mesh. (If HORZREGS = 8, then 8 distances should be given here.)

5. VERTDIST: The thickness, h, of each vertical region down the mesh.

6. HORZNODE: The number of nodes, n, within each horizontal region across the mesh.

7. VERTNODE: The number of nodes, m, within each vertical region down the mesh.

8. MSWEEP1: Indicates one of the four corners at which the odd-numbered mesh sweeps are to start. (TOP LEFT, TOP RIGHT, BOTTOM LEFT, BOTTOM RIGHT) It also indicates whether the sweep is to move across the rows or along the columns. (ROW or COL respectively)

9. MSWEEP2: Indicates the corner at which the even-numbered mesh sweeps are to start. It also indicates whether the sweep is to move across the rows or along the columns.

3.6 ***NAMES*** Section

The only input section which uses "name-expressions" is the ***NAMES*** section. This is one of those five remaining sections which are allowed to appear in any order in the input stream. The purpose of this section is simply to give names and numbers to the various materials in the problem. Before execution each material should be assigned a number and a name through the use of these "name-expressions." During execution the user will then be able to refer to these various materials in the printout by name or number. However in the other input sections, the various physical properties should be assigned to each material only by number, not name.

Only one keyword is presently acceptable as an expression-name in this section: <u>MAT</u>. Of course, it may be written as <u>MAT</u>ERIAL NO, since only the first three letters are looked at. The general form of the name-expression is

$$\text{MATERIAL NO} = \begin{Bmatrix} i \\ \underline{0} \end{Bmatrix} \quad \text{FOR} \quad \begin{Bmatrix} \text{material-name} \\ \underline{\text{bbbbbbbb}} \end{Bmatrix} . \tag{3-8}$$

where "i" stands for a positive integer and "b" stands for a blank. The underlined values are again the defaults assumed if no expression is given. It is illegal, of course, to write the expression and leave out the integer or the material-name. In this type of expression the word, FOR, must be present. Only the first eight characters of the material-name are used in the computer printout. A typical example might look like:

$$\text{MATERIAL NO} = 3 \quad \text{FOR} \quad \text{CONCRETE} . \tag{3-9}$$

3.7 ***BOUNDARY*** Section

The purpose of this input section is to assign boundary conditions to the outer edges of the mesh assembly. As was stated earlier, the boundary conditions may be specified by a fixed surface temperature, T_b, or a relation of the form:

$$-\frac{\partial T}{\partial n_b} = a(T_b - T_{ext}) = \frac{h}{k_b}(T_b - T_{ext}) \qquad (3\text{-}10)$$

where 1/a is the boundary layer thickness called the gradient distance (ft, in, etc.). The two parameters, k_b and h, are the conductivity of the surface material and the heat transfer rate across the surface (BTU/ft^2hr^oF) respectively. Other types of boundary conditions, such as radiation, may be used in the transient code by letting the boundary conditions be a function of temperature. (See Section 2.5.) The boundary edges of the assembly should be (1) insulated, (2) given a fixed surface temperature, or (3) assigned an external temperature and boundary layer thickness. In steady-state problems these boundary conditions must remain fixed, but in transient problems the T_b, T_{ext}, or 1/a values may change at time steps specified by the user. All of this is done by using the appropriate parameters shown in Table 3.6. The notation used in Table 3.6 is described in Section 3.2. The underlined letters within certain words are the only ones looked at and the rest are extraneous. Four parameters are acceptable to both codes and two additional parameters are available in the transient code. The "f" represents any floating-point number, and the Type-1 Integer-expressions are defined in Section 3.2. Notice that some of the location-names have been divided into two separate parts which are to be joined

together by a period. Any of the items in the first part may be joined with any of the items in the second part (TOP.SIDE, TOP.NODES, LHS.REGIONS, etc.).

Table 3.6

BOUNDARY

[INSULATED / EDGE TEMP / EXT TEMP / GRAD DIST] = {f} [units] FOR {I/O.FUNCTION / {TOP / RHS / BOT / LHS} . {SIDE / REGIONS / NODES}} [Type-1 Integer-expressions] {, / .}

Transient Only:

[TEMP CHANGE / DIST CHANGE] = {f} [units] FOR {I/O.FUNCTION / ALL.TIME / TIM.STEPS} [Type-1 Integer-expressions] {, / .}

The expressions in Table 3.6 are somewhat self-explanatory although the location-names need a detailed explanation. The following list discusses the six keywords used as expression-names:

1. INSULATED: Indicates that certain portions of the boundary are to be insulated from the external environment. The "f" value should be zero in this case.

2. EDGE TEMP: The surface temperature, T_b, of the boundary.

3. EXT TEMP: The external temperature, T_{ext}, of the outside environment. If EXT TEMP is used then a GRAD DIST must also be given.

4. GRAD DIST: The boundary layer thickness, 1/a, in feet, inches, etc. If GRAD DIST is used then an EXT TEMP must also be given.

5. TEMP CHANGE: Indicates that the outside temperature, T_b or T_{ext}, is to change at the specified time steps. A subprogram will

be called with the "f" value as its argument so that the T_b or T_{ext} can be changed for the desired boundary nodes. (See Section 2.5.)

6. DIST CHANGE: Indicates that the boundary layer thickness, 1/a, is to change at the specified time steps. A subprogram will be called with the "f" value as its argument so that 1/a can be changed for the desired boundary nodes. (See Section 2.5.)

To explain the location-names let us consider the four boundaries of the assembly shown below:

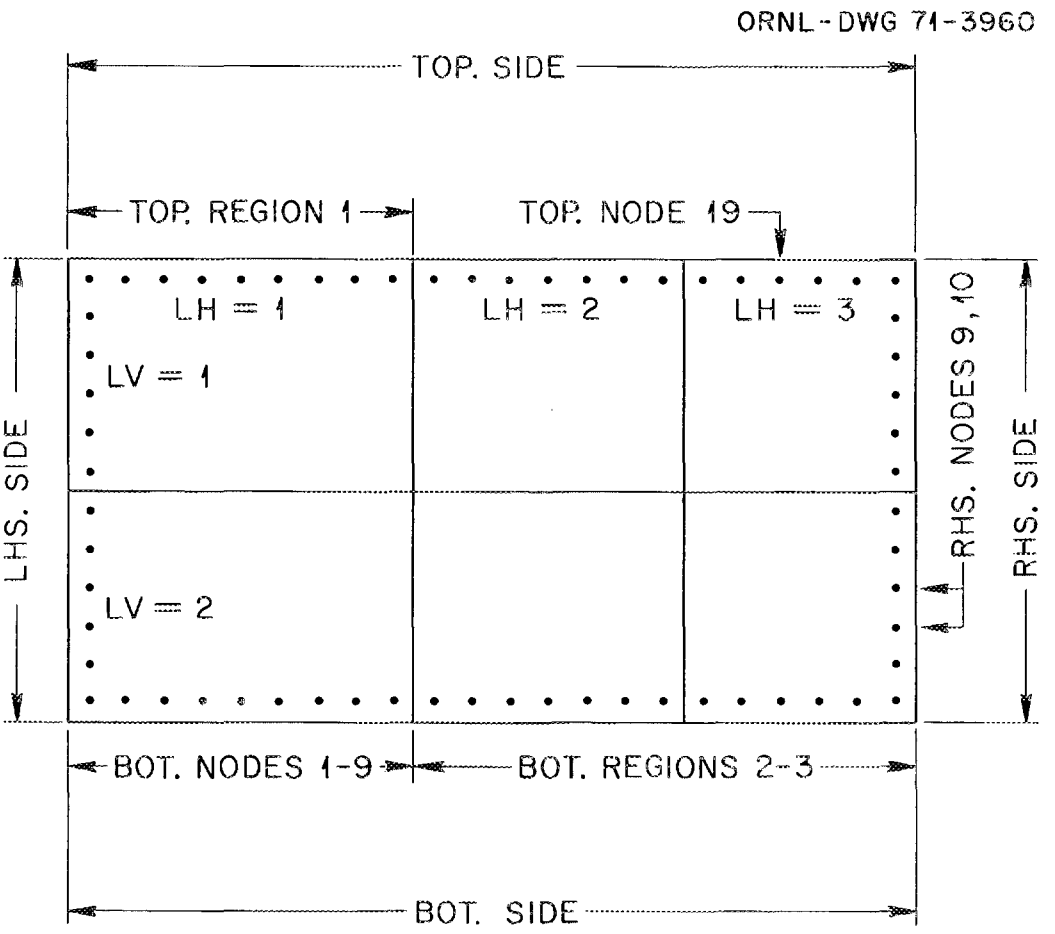


Figure 3.3

The location-names allow us to refer to the nodes on the boundary in several different ways. The first half of the location-name specifies one of the four sides and the second half specifies the affected nodes on that side. Table 3.7 discusses these two halves in detail.

Table 3.7

1.	TOP:	Indicates the top boundary of the assembly ($z = 0$ line)
2.	RHS:	Indicates the right-hand-side boundary of the assembly ($r = r_{max}$ line)
3.	BOT:	Indicates the bottom boundary of the assembly ($z = z_{max}$ line)
4.	LHS:	Indicates the left-hand-side boundary of the assembly ($r = 0$ line)
1.	SIDE:	Indicates that all the boundary nodes along one complete side are to be used. An integer-expression should not follow the location-name.
2.	REGIONS:	Indicates that all the boundary nodes within one or more regions are to be used. The regions are numbered either horizontally or vertically depending on the side involved. An integer-expression should follow the location-name to specify these regions.
3.	NODES:	Indicates that certain individual boundary nodes are to be used. The nodes are numbered either horizontally or vertically depending on the side involved. An integer-expression should follow the location-name to specify the nodes that are affected.

Three location-names have not been included in Table 3.7: I/O.FUNCTION, ALL.TIME, and TIM.STEPS. This is because they are not made up from two separate halves joined by a period. ALL.TIME and TIM.STEPS, as their names imply, simply indicate the time steps at which the boundary conditions are to change. If ALL.TIME is used then the boundary conditions change at

every time step and no integer-expression is needed. However, if TIM.STEPS is used then an integer-expression should follow it to indicate the specific time steps at which the changes are to take place. Section 2.5 and Appendix E contain a description of the two subprograms, TOBARF and ARATEF, which are called to change the boundary conditions.

The I/O.FUNCTION keyword has been provided to allow more flexibility in reading input data. Perhaps the user wishes to read boundary temperatures from a library of tapes written in an unusual format. He can do so by writing a FORTRAN function called BOUFUN to read the tapes and assign the external temperatures to the proper boundary nodes. Then during execution he includes the following card in his input stream:

<u>EXT</u> TEMP = 1 CALL TO <u>I/O.FUN</u>CTION WITH LUNIT 15 . (3-11)

Notice the use of extra commentary within the expression to make it more readable. The I/O.FUNCTION, BOUFUN, will be called one time with the logical unit number 15 as an argument. A skeleton BOUFUN function has already been written so that the user will already have the calling sequence set up.

Three other input sections, yet to be described, also allow the I/O. FUNCTION keyword for user-controlled input. There is a different function name (like BOUFUN) for each type of parameter. For example, the conductivity expression will call CONFUN whenever I/O.FUNCTION is used as a location name. A list of the various parameters and their I/O.FUNCTION is given in Table 3.8.

Table 3.8

Parameter Name	Input Section	I/O.FUNCTION Called
INSULATED	***BOUNDARY***	BOUFUN
EDGE TEMP	***BOUNDARY***	BOUFUN
EXT TEMP	***BOUNDARY***	BOUFUN
GRAD DIST	***BOUNDARY***	BOUFUN
TEMP CHANGE	***BOUNDARY***	BOUFUN
DIST CHANGE	***BOUNDARY***	BOUFUN
MATERIAL NO	***MATERIALS***	MATFUN
CONDUCTIVITY	***MATERIALS***	CONFUN
DENSITY	***MATERIALS***	SPWFUN
SPHEAT	***MATERIALS***	SPHFUN
MELTING TEMP	***MATERIALS***	MELFUN
LATENT HEAT	***MATERIALS***	LATFUN
HEAT SOURCE	***MATERIALS***	HEAFUN
POWER LOAD	***MATERIALS***	POWFUN
TEMP	***TEMPERATURES***	TEMFUN
AVG TEMP	***TEMPERATURES***	AVGFUN
FIXED TEMP	***TEMPERATURES***	FIXFUN
MAX TEMP	***TEMPERATURES***	FIXFUN
TIME STEP INCR	***TIME***	TIMFUN
TEMP LIST FREQ	***TIME***	LISFUN

For most applications, however, Table 3.7 is quite sufficient for describing the boundary conditions.

3.8 ***MATERIALS*** Section

The ***MATERIALS*** input section is the largest and most complicated of all the input sections. Its purpose is threefold: first, it assigns materials to each of the regions within the mesh assembly; second, it assigns the physical properties to each of these materials; and third, it assigns any heat sources present to their respective areas within the mesh. Table 3.9 shows the eight keyword parameters which are designed to accomplish these three purposes. The first parameter assigns materials to regions, the next five assign properties to materials, and the last two assign heat sources to areas. Those items which are preceded with an asterisk should only be used in the transient code. All the rest are acceptable to both codes.

In some of these "for-expressions" the numeric value has been specified as either a floating-point number, "f," or as the specific constant, 9999999. When this specific constant is used, it indicates that the particular parameter involved, such as conductivity, is to be calculated by an external function during the actual solution of the heat transfer problem. In this way the user can indicate that he wants the thermal conductivity to be a function of temperature for certain materials. See Section 2.5 for a detailed description of temperature, position, and time-dependent functions. Table 2.1 and Appendix E list the various external functions which are called when 9999999 is used. (Notice that there are seven 9's.) Do not confuse the use of 9999999 with the use of the location-name, I/O.FUNCTION. Section 3.7 describes how I/O.FUNCTIONs may be used (during input only) to read unusually formatted input data, such as conductivity values, and assign them to the proper materials.

Table 3.9

MATERIALS

[MATERIAL NO]	= {i}	FOR	I/O.FUNCTION ALL.MESH COL.REGIONS ROW.REGIONS INT.REGIONS	[Type-1 Integer-expression]	{, .}
			R,Z.REGIONS Y,Z.REGIONS R,T.REGIONS	[Type-2 Integer-expression]	
[CONDUCTIVITY *SPHEAT *DENSITY]	= {f 9999999} [units]	FOR	I/O.FUNCTION ALL.MESH MAT.NOS *LIQ.MATERIALS	[Type-1 Integer-expression]	{, .}
[*MELTING TEMP *LATENT HEAT]	= {f} [units]	FOR	I/O.FUNCTION ALL.MESH MAT.NOS	[Type-1 Integer-expression]	{, .}
[HEAT SOURCE POWER LOAD]	= {f 9999999} [units]	FOR	I/O.FUNCTION ALL.MESH COL / ROW / INT . REGIONS / NODES	[Type-1 Integer-expressions]	{, .}
			R,Z / Y,Z / R,T . REGIONS / NODES	[Type-2 Integer-expressions]	

*Note: Those keywords with an asterisk are only allowed in the Transient code.

The following list discusses the eight keywords which are used as expression-names in the preceding table:

1. MATERIAL NO: The material number (integer only) of the material which is to be assigned to various regions within the mesh. The location-name and integer-expressions which follow indicate the particular regions involved. Every region must be assigned a given material.

2. CONDUCTIVITY: The thermal conductivity of those materials specified by the location-name and integer-expressions which follow. English units = B/FH'F.

3. SPHEAT: The specific heat of those materials specified by the location-name and integer-expressions which follow. English units = B/P'F.

4. DENSITY: The density of those materials specified by the location-name and integer-expressions which follow. English units = P/FFF.

5. MELTING TEMP: The melting temperature of those materials specified by the location-name and integer-expressions which follow. If a material is given a MELTING TEMP it must also be given a LATENT HEAT. Its other physical properties should also be given for both the liquid and solid state. English units = 'F.

6. LATENT HEAT: The latent heat of fusion of those materials specified by the location-name and integer-expressions which follow. If a material is given a LATENT HEAT it must also be given a MELTING TEMP. Its other physical properties should also be given for both the liquid and solid state. English units = B/P.

7. HEAT SOURCE: The heat source value to be assigned to those areas indicated by the location-name and integer-expressions which follow. This value is placed in every node within the specified areas. English units = B/HFFF.

8. POWER LOAD: Same as HEAT SOURCE except the value given here is not per unit volume, and thus it applies to the indicated mesh, region, or node as a whole. Before this value is placed into every node within the specified area, it is divided by the volume of the indicated mesh, region, or node. English units = B/H. (MKS units = Watts) After division by the appropriate volume, its units = B/HFFF.

As indicated above, the POWER LOAD parameter has been included to allow the user to specify the total heat per unit time (B/H.) generated within a given region or node (or the whole mesh). The code will then automatically convert this power load value into heat per unit volume per unit time (B/HFFF.) so that each affected node may be assigned the proper heat source value. If several regions are specified in one POWER LOAD expression, such as:

POWER LOAD = 1500 W. FOR R,Z.REGIONS (5,1) (6,2) (3,2) . (3-12)

it is assumed that the power load value applies to each independently, and not to all the regions as a whole.

Some of the location-names in Table 3.9 are self-explanatory but those involving REGIONS and NODES need a detailed explanation. As in the preceding section, some of these location-names consist of two halves joined together by a period. Any items in the left half may be joined with any items in the right half. For example: COL.REGIONS, COL.NODES, INT.REGIONS, R,Z.REGIONS, R,T.NODES, etc. As shown in Table 3.9 certain location-names are used only with Type-1 Integer-expressions while others are used only with Type-2 Integer-expressions. Perhaps the best way to explain these REGION and NODE location-names is with an example. Let us consider the mesh assembly shown in Fig. 3.4. It is very easy to identify a region or node by its row and column numbers or by its internal number. When a whole column (or row) or regions is to be considered, then the location-name, COL.REGION (or ROW.REGION), should be used; likewise for a whole column or row of nodes. Note that whenever a region is specified, all the nodes within that region are affected. The following list discusses

ORNL-DWG 71-3961

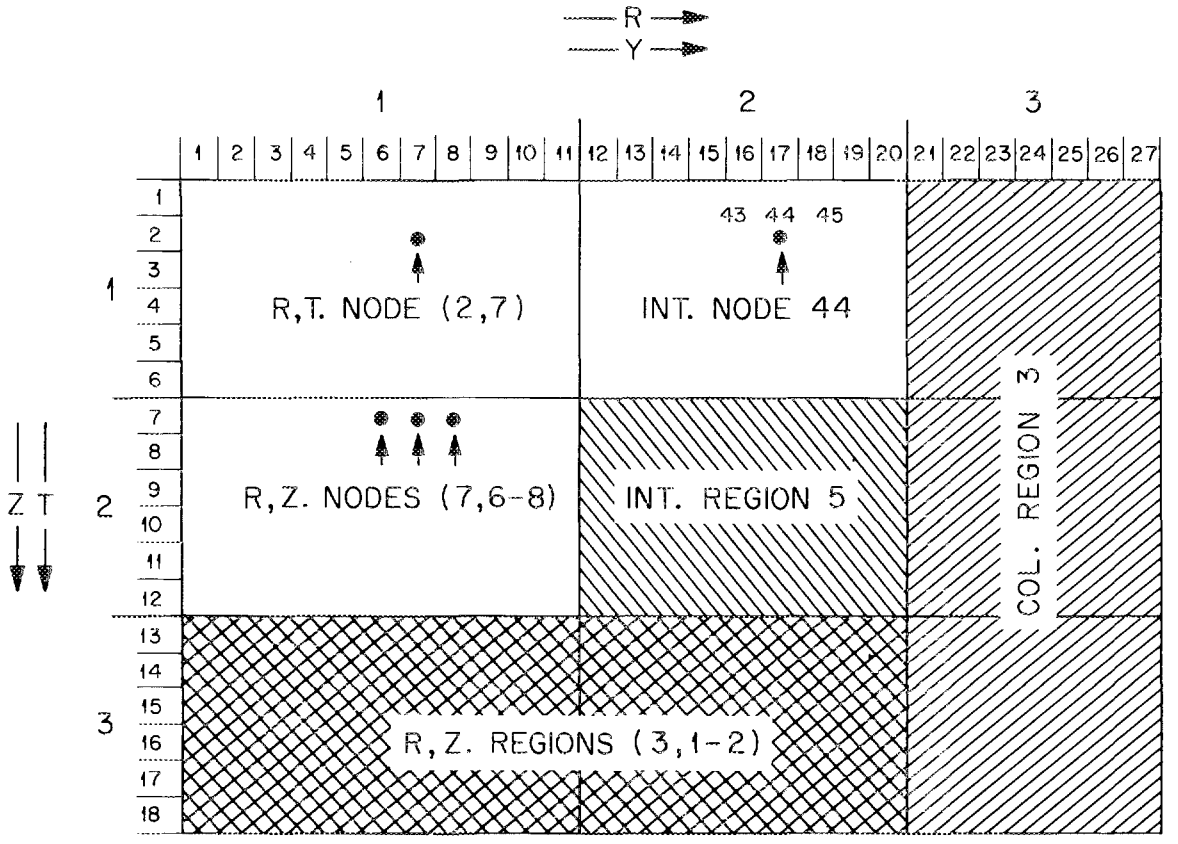


Figure 3.4

all of the location-names used in Table 3.9. Sample integers follow some of the location-names to enhance the explanation.

1. I/O.FUNCTION: Indicates that a user-written input function is to be called to supply the input. See Section 3.7 for a detailed explanation.

2. ALL.MESH: Indicates that the complete mesh (every node, region, or material) is to be used.

3. MAT.NOS 2 4 5: Indicates that material numbers 2, 4, and 5 are to be used. If melting is considered, then MAT.NOS represents the solid state of these materials.

4. LIQ.MATERIALS 4 5: Indicates that materials 4 and 5 are to be used in their liquid state. In this way conductivity values, etc., may be specified for materials in the liquid state as well as the solid state.

5. COL.REGIONS 3 5: (or NODES) Indicates that the third and fifth column of regions (or nodes) are to be used. These columns are assumed to extend from the top of the mesh all the way to the bottom.

6. ROW.REGIONS 2 4: (or NODES) Indicates that the second and fourth row of regions (or nodes) are to be used. These rows are assumed to extend from the leftmost boundary to the rightmost boundary.

7. INT.REGIONS 5 7: (or NODES) Indicates that the two regions (or nodes) whose internal numbers are 5 and 7 are to be used. These internal numbers increase through the mesh in normal reading order, from left to right and top to bottom. See Section 2.2 for a further description.

8. R,Z.REGION (3,1): (or NODE) Indicates that the region (or node) whose row number is 3 and column number is 1 is to be used. Parenthetical integer-expressions (Type-2) are to be used only when the first half of the location-name is R,Z or Y,Z or R,T.

9. Y,Z.REGION (3,1): (or NODE) Same as R,Z.REGION (3,1). The Y-Z half has been provided for use with the y-z (same as x-y) geometry.

10. R,T.REGION (3,1): (or NODE) Same as R,Z.REGION (3,1). The R,T half has been provided for use with the r-θ geometry.

One time-saving feature of these two codes involves the concept of "painting," or overlaying one material with another. When the user is preparing his mesh assembly, he may find that one material, say steel, appears all through his assembly in several noncontiguous regions. To define every little region of steel independently would be tedious. However, by first putting steel everywhere (ALL.MESH) and then painting the less frequent materials over the steel (R,Z.REGIONS ..., etc.), he can make the job a lot easier. He must be careful that the cards describing the less frequent materials (MATERIAL NO cards) come after the one

describing the steel. This same procedure may also be used in other places such as the description of initial temperature distributions.

In preparing ***MATERIALS*** section input, it might be helpful at this point to refer to the example in Figs. 3.1 and 3.2.

3.9 ***TEMPERATURES*** Section

The purpose of this input section, as its name implies, is to specify temperatures of one sort or another. The primary purpose is to specify the initial temperature distribution for all nodes in the mesh, whether it be a starting guess for a steady-state case or the known temperatures at the start of a transient case. In addition, three other parameters are available in this input section to allow the user to (1) fix any temperatures within the mesh at a constant value, (2) specify maximum allowable temperature for any given nodes in a transient problem, and (3) average several groups of nodal temperatures for printout purposes. These three input parameters plus the initial temperature parameter are shown in Table 3.10. Again, a preceding asterisk indicates items acceptable only to the transient code.

Table 3.10

TEMPERATURES

```
[ TEMP          ]
[ FIXED TEMP    ]   = {f} [units] FOR { I/O.FUNCTION }   [Type-1 Integer-]  {
[ *MAX TEMP     ]                     { ALL.MESH     }   [expressions    ]
                                      { {COL} . {REGIONS} }
                                      { {ROW}   {NODES}   }
                                      { {INT}             }
                                      { {R,Z} . {REGIONS} }   [Type-2 Integer-]
                                      { {Y,Z} . {NODES}   }   [expressions    ]
                                      { {R,T}             }

[ AVG TEMP GROUP ]  = {i}          FOR { I/O.FUNCTION }   [Type-1 Integer-]  {,}
                                       { ALL.MESH     }   [expressions    ]  {.}
                                       { {COL} . {REGIONS} }
                                       { {ROW}   {NODES}   }
                                       { {INT}             }
                                       { {R,Z} . {REGIONS} }   [Type-2 Integer-]
                                       { {Y,Z} . {NODES}   }   [expressions    ]
                                       { {R,T}             }
```

*Note: The keyword with an asterisk is only allowed in the Transient code.

These four parameters are described in the list below.

1. TEMP: The initial temperature to be used at the start of either the steady-state or transient codes. Every node in the mesh must eventually be assigned an initial temperature. The location-name and integer-expressions which follow indicate the particular nodes to be assigned this temperature. If a whole region is specified, then every node in that region is assigned this initial temperature value.

2. FIXED TEMP: The constant temperature value to be assigned to those nodes specified by the location-name and integer-expressions which follow. Each of these particular nodes remains fixed at this constant temperature throughout the problem. Any of the nodes within the mesh may be fixed.

3. MAX TEMP: The maximum temperature to be allowed for those nodes specified by the location-name and integer-expressions which follow. This parameter is valid only in the transient code. An error message is printed the first time each of these nodes exceeds the maximum temperature. If ALL.MESH is used then every node in the mesh is checked against this maximum temperature.

4. AVG TEMP GROUP: Indicates that one or more nodes are to have their temperatures averaged and printed out every time a line of iterative printout occurs. The particular nodes involved are indicated by the location-name and integer-expressions which follow. Up to six different groups of these node sets may be specified, thus resulting in six different average temperatures printed out.

The user may input as many TEMP cards as necessary to describe his initial temperatures. However, there is an upper limit of 250 on the number of different fixed temperature values or maximum temperature values that can be specified for one problem.

In the AVG expression shown in Table 3.10, the numeric value should be an integer (1 to 6) specifying a group number for the set of nodal temperatures being averaged. As many as six different groups may be chosen. This option enables the user to quickly follow the temperature changes at

important areas within his mesh assembly. To look at the temperatures of the outer two corners in Fig. 3.4 we would use an AVG expression like:

$$\text{AVG TEMP GROUP} = \begin{array}{l} 1 \text{ FOR R,Z.NODES } (1\text{-}2,26\text{-}27) \ , \\ 2 \text{ FOR R,Z.NODES } (17\text{-}18,26\text{-}27) \ . \end{array} \qquad (3\text{-}13)$$

Note that the temperature averaging is done over the four nodes in each corner. There is no averaging of temperatures over time.

The FIXED TEMP parameter has proven useful in several different ways. For example, it was quite helpful in considering structures with internal cooling pipes held at constant temperatures. At present, it is the best way to consider structures with irregular boundaries or internal boundaries. Suppose we had a mesh assembly with the upper-right corner cut out as shown in Fig. 3.5. We can treat the inside surfaces of the cutout area as an external boundary by fixing the temperature of the corner nodes at the external temperature, T_{ext}, and setting the conductivity of these nodes to $h \cdot \Delta x$. Since this technique is rather messy and time-consuming, we plan to improve these two codes in the future so that irregular boundaries may be handled more easily.

The location-names used in Table 3.10 are identical to those associated with the HEAT SOURCE and POWER LOAD parameters in Table 3.9. Thus, Section 3.8 already contains a complete description of these names and they need not be repeated here. Section 3.12 tells how to save a final steady-state temperature distribution on tape, and then start a transient problem using this distribution as initial temperatures at time = 0.

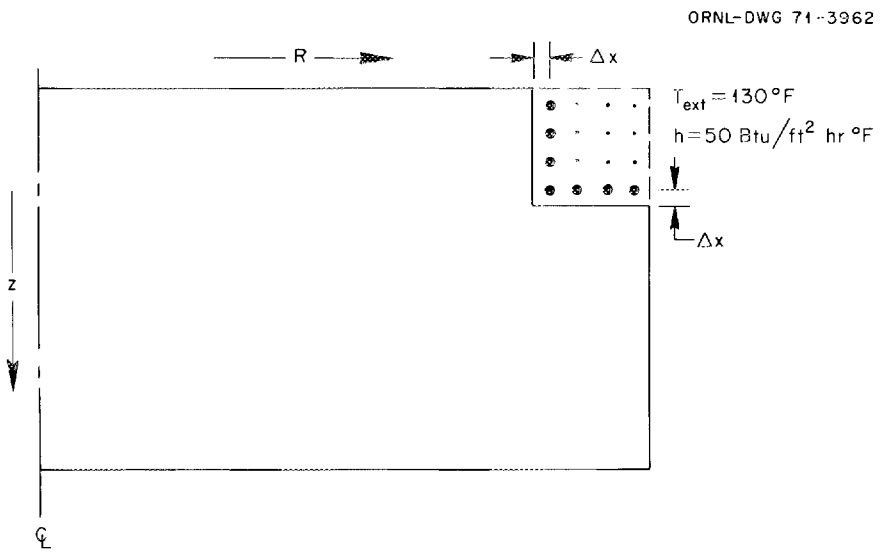


Figure 3.5

3.10 ***TIME*** Section

The ***TIME*** input section is only to be used with transient problems since no passage of time occurs in steady state. There are only two pieces of information needed here. The first is the time step increment and the second is the frequency of printing temperature distributions. The number of time steps to be used in the problem has already been specified in the ***HEADER*** input section. Thus only the length (secs, mins, etc.) of these steps needs to be given here. As far as printout is concerned, it usually is not necessary for the full temperature array to be printed at every time step, and thus the TEMP LIST FREQ parameter has been provided to indicate at which time steps the full array should be printed. To keep a running check of certain key temperatures the AVG TEMP GROUP parameter (Section 3.9) may be used. Table 3.11 shows both of these parameters along with their location names.

Table 3.11

TIME

[TIME STEP INCR]	=	{f} [units]	FOR	I/O.FUNCTION / ALL.TIME / TIM.STEPS	[Type-1 Integer-expression] {, / .}
[TEMP LIST FREQ]	=	{i}	FOR	I/O.FUNCTION / ALL.TIME / TIM.STEPS	[Type-1 Integer-expression] {, / .}

Notice that the TIME STEP INCR expects a floating-point number, "f" (secs, mins, etc.), whereas the TEMP LIST FREQ expects a positive integer. The rest of these two expressions are identical. The list below describes the expression-names and location-names used above:

1. TIME STEP INCR: The length of time, Δt, for those time steps given in the location-name and integer-expressions which follow. All time steps must be given a Δt value. Of course, the Δt value may vary from one step to another.

2. TEMP LIST FREQ: The frequency, n, at which the full temperature array is to be printed. Although this particular frequency value applies only to the TIM.STEP range which follows, every time step may be assigned some frequency value. If n = 3 for TIM.STEPS 14-28, then temperatures will be printed at time steps 16, 19, 22, 25, and 28. Thus a printout will occur every n steps within the specified range.

3. I/O.FUNCTION: Indicates that a user-written input function is to be called to supply the input. See Section 3.7 for a detailed explanation.

4. ALL.TIME: Indicates that all the time steps are to be used (1 though NTIMSTEP). No integer-expression can follow this keyword.

5. TIM.STEPS 1-80: Indicates that the first through the eightieth time steps are to be used. An integer-expression must follow this keyword to specify the actual time steps involved.

In many heat transfer problems the temperatures change very quickly at the beginning of the problem and then settle down later on. In such problems it is advantageous to use variable length time steps so that small time increments may be used when temperatures are changing quite rapidly and longer increments may be used when they settle down. This is done by specifying different TIME STEP INCRs for different TIM.STEPS as shown in the example below.

```
TIME STEP INCR  =  10 S.  FOR  TIM.STEPS  1-30   ,
                   20 S.  FOR  TIM.STEPS 31-60   ,          (3-14)
                   30 S.  FOR  TIM.STEPS 61-90   ,
                   40 S.  FOR  TIM.STEPS 91-120  .
```

In Section 3.8 the "painting" (overlaying) technique was described for MATERIAL NOs. This same technique may be used here in specifying either TIME STEP INCRs or TEMP LIST FREQs. For example, to print temperatures every 10 steps, except during the steps 50-100 when temperatures are needed every 5 steps, we could write:

```
TEMP LIST FREQ  =  10  FOR  ALL.TIME  ,
                    5  FOR  TIM.STEPS 50-100  .          (3-15)
```

The later value always overlays the earlier one for the time steps indicated

3.11 Parametric Studies

In many problems that involve heat transfer, especially during the early phases of design, a user may wish to know what effect certain variations in the system might have on his final solution. Such a study would involve running the same problem several times with certain parameters being changed each time. For example, during the preliminary design of a cylindrical shipping cask, the designer may wish to use a heat transfer code to find the steady-state temperatures inside the cask. However, he may not know the exact conductivity of certain materials within the cask nor the exact boundary conditions on the outer surfaces. Because of these uncertainties, he could choose three or four representative values for his conductivity and three or four values for the boundary conditions, and then run nine or ten independent problems with different combinations of these values to see the effect on the final temperatures. The preparation of the input and the running time for all these problems would be much greater than that for a single problem. When using ORTHIS and ORTHAT, all this ("and all that") is not necessary. All the user needs to do is prepare the complete input for his initial problem and then follow it with as many additional cases as he would like. These additional cases need not contain a complete description of the whole problem again, but instead need only the description of those parameters which are changing. As each new case is encountered the final temperature distribution from the one before will be used as a starting guess, thereby decreasing the running time significantly. The only cards required in each additional case are the title cards and the ***ENDATA*** card. Figure 3.6 gives an

example of the input cards needed to run an initial problem plus three additional cases.

There are many different ways in which the input may be stacked so that one case can take advantage of the previous case just run. Thus the overall advantages of stacking cases are that certain parameters may be varied from one case to the next and the corresponding problems can then be solved in much less time than the original problem.

```
ORTHIS STEADY-STATE TITLE CARD FOR INITIAL PROBLEM
                                ***HEADER***
     INITIALZ = YES  ...
     ...
                                 ***MESH***
     SYSTEM = R,Z  ...
     .
     .
     .
                               ***BOUNDARY***
     GRAD DIST = 0.25 F.  FOR  RHS.SIDE .
     .
     .
     .
                               ***MATERIALS***
     CONDUCTIVITY = 10.0 B/FH'F.  FOR  MAT.NOS 1 .
     .
     .
     .
                                ***ENDATA***

ORTHIS STEADY-STATE TITLE CARD FOR ADDITIONAL CASE 1
     ***BOUNDARY***     GRAD DIST = 0.30 F.  FOR  RHS.SIDE .
     ***MATERIALS***    CONDUCTIVITY = 15.0 B/FH'F.  FOR  MAT.NOS 1 .
     ***ENDATA***

ORTHIS STEADY-STATE TITLE CARD FOR ADDITIONAL CASE 2
     ***BOUNDARY***     GRAD DIST = 0.35 F.  FOR  RHS.SIDE .
     ***MATERIALS***    CONDUCTIVITY = 20.0 B/FH'F.  FOR  MAT.NOS 1 .
     ***ENDATA***

ORTHIS STEADY-STATE TITLE CARD FOR ADDITIONAL CASE 3
     ***BOUNDARY***     GRAD DIST = 0.40 F.  FOR  RHS.SIDE .
     ***MATERIALS***    CONDUCTIVITY = 25.0 B/FH'F.  FOR  MAT.NOS 1 .
     ***ENDATA***
```

Figure 3.6

3.12 Restart Facilities

In using these two heat transfer codes, ORTHIS and ORTHAT, it is sometimes necessary to start a machine run with a temperature distribution that has been previously calculated. This ability to restart a problem from a temperature distribution that has already been saved on tape or disk has proven useful in several different ways. If a big problem cannot be completed in one machine run due to time limitations, the last temperatures calculated may be saved on tape for beginning the next run. Also the initial temperatures (t = 0) for a transient problem may come from steady-state calculations already done. In this case the final steady-state temperatures should be saved on tape or disk. Once in a while, starting temperature distributions come in from outside sources in a different format than normally used in ORTHIS and ORTHAT. In this case, the I/O.FUNCTION for temperatures, TEMFUN, can be easily altered to accept the new format. It would be a good idea for the reader to refer to Section 3.7 and Table 3.8 for an explanation of the I/O.FUNCTION keyword.

It is very simple to save temperatures at the end of a machine run on some logical output unit. The following two parameters in the ***HEADER*** section indicate whether to save temperatures, and if so what output device to save them on:

SAVETEMP = YES SAVEUNIT = 2 (3-16)

In this example, the final temperatures will be written in binary onto the device whose logical unit number is 2. It is the user's responsibility to supply an FT02F001 DD control card with a DCB field something like:

DCB=(RECFM=V,LRECL=3600,BLKSIZE=3604) (3-17)

See ORNL Programmer's Manual, Sec. 9.

To restore temperatures that have been saved according to the above procedure, the user simply includes the following card:

TEMP = 1 CALL TO I/O.FUNCTION WITH LUNIT 2 . (3-18)

in the ***TEMPERATURES*** input section. This causes the subroutine, TEMFUN, to be called so that it can read the temperatures from logical unit 2. If logical unit 2 contains a tape with temperatures written in a format different from that described above, then TEMFUN will have to be changed to accept the new format. Appendix F contains a listing of the present TEMFUN subroutine.

3.13 Maximum Allowable Sizes

Within these two codes, ORTHIS and ORTHAT, are numerous arrays which have been dimensioned quite large so that big problems may be handled. The upper limits currently being used are given in Table 3.12. Since these limits do not use all the memory available on ORNL's computers (Steady-state ≈ 700 K; Transient ≈ 1024 K), they may be increased if necessary. Likewise they may be decreased for use with a smaller machine.

Table 3.12

Variable Name	Current Upper Limit	Description
IMAX	1000	Maximum number of vertical nodes (rows)
JMAX	1000	Maximum number of horizontal nodes (columns)
KMAX	10000	Maximum number of all nodes (I*J)
LVMAX	50	Maximum number of vertical regions (rows)
LHMAX	50	Maximum number of horizontal regions (columns)
LMAX	2500	Maximum number of all regions (LVMAX*LHMAX)
MNMATL	300	Maximum number of materials
MNITR	1500	Maximum number of steady-state iterations
MNITR	5000	Maximum number of transient time steps

4. OUTPUT DESCRIPTION

4.1 General

The best way to describe the output from these two codes, ORTHIS and ORTHAT, is to use the printout from a sample problem. This sample problem involves the use of both the steady-state and the transient codes. However, since the output format from both codes is almost identical, only the transient printout will be used (except in Section 4.4).

The problem consists of a cylindrical shipping cask which is subjected to a fire at 1475°F. Before the fire begins, the steady-state temperatures must be calculated and saved on tape. Then these temperatures can be used as the initial temperatures when the fire actually begins in the transient problem. The shipping cask is assumed to be insulated on the bottom so that only the sides and the top are exposed to the fire. The cask contains an internal source at its center generating heat at the rate of 1200 watts. Figure 4.1 shows a cross-sectional view of the shipping cask with the centerline (r = 0) at the leftmost edge. Temperatures will be calculated and printed out for the first thirty minutes of the fire.

The description of the computer printout from this sample problem has been divided into four parts, Sections 4.2-4.5. The first part is simply a printout of the input data read by the code. The second is a picture of the mesh assembly which shows where each material is located within the assembly. In addition, the physical properties of these materials are listed, and the nodal and regional distances are given in a table. The third part consists of a single line of printout at every

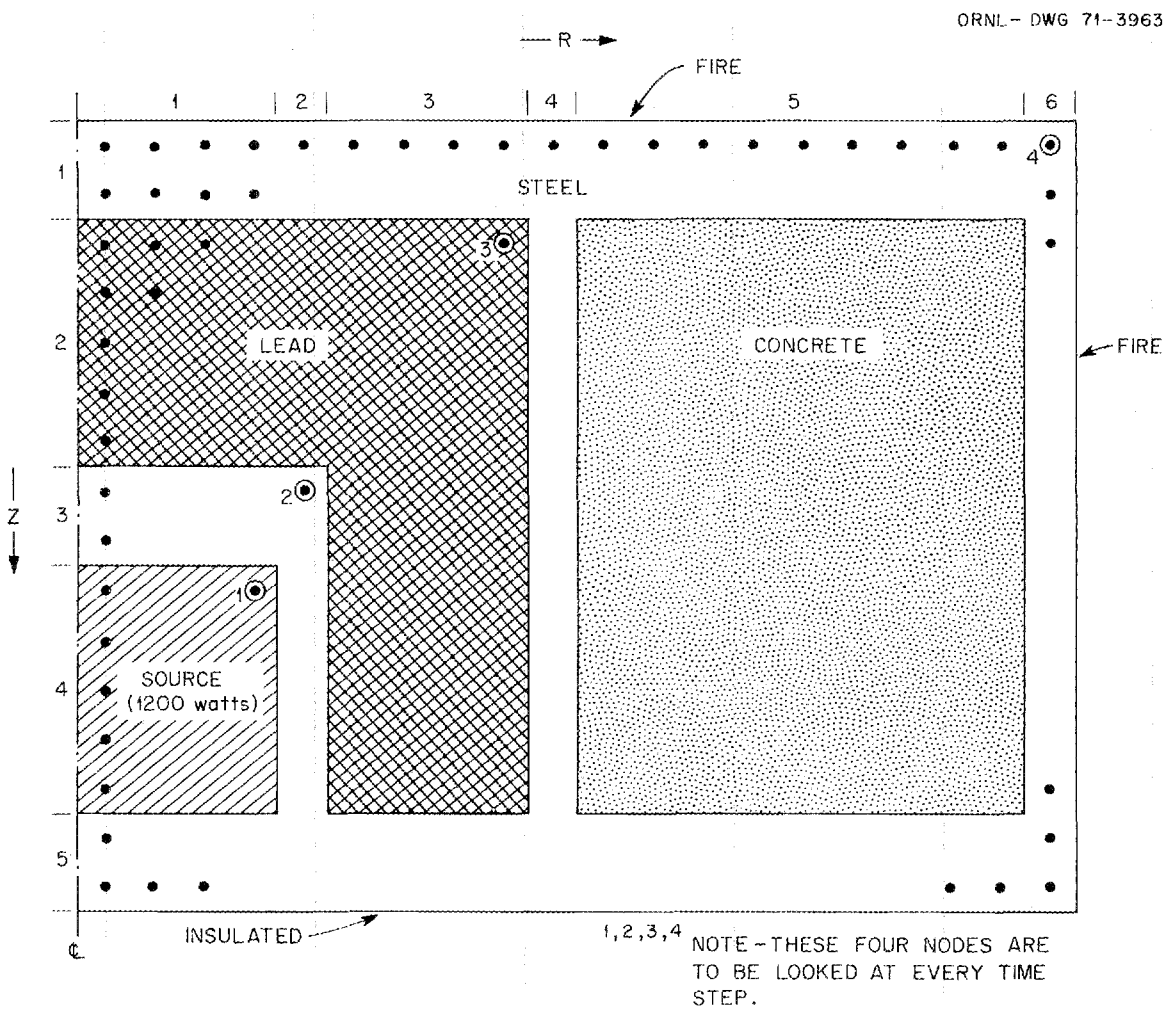

ORNL-DWG 71-3963
R
FIRE
1
2
3
4
5
6
STEEL
LEAD
CONCRETE
FIRE
Z
SOURCE
(1200 watts)
INSULATED
℄
1,2,3,4 NOTE-THESE FOUR NODES ARE TO BE LOOKED AT EVERY TIME STEP.

Figure 4.1

iteration. This line is a summary of what is happening within the mesh at each iteration. In the case of the steady-state problem, it indicates how quickly convergence is being reached, whereas in a transient problem it indicates how certain temperatures are varying with time. Since these two iterative summaries are different, both are included in Section 4.4. The fourth part of the printout describing the full temperature array is in Section 4.5. Several different steady-state plots were generated to show the temperature distributions and profiles after convergence was reached. These are shown in Section 4.6. The procedure for writing the final steady-state temperatures onto tape has already been described in Section 3.12.

4.2 Printout of Input Data

The first page of printout always contains the problem title as punched on the user's title cards. Following this first page comes the printout of the input data, by section. There are seven different input sections which may be used in the transient code; six in the steady-state code. Figure 4.2 shows the actual cards which were used as input for our sample transient problem. These may be compared with the computer printout shown in Fig. 4.3 of this same data. Notice that each input expression is printed on a different line even though several expressions may have been punched on the same card. Numeric values within some expressions may be printed twice to show the conversion from the input units to the standard internal units specified in the ***HEADER*** section. In this particular shipping cask problem we have specified that the internal units for time are to be minutes. As was explained in Section 3.3, the letter "M" was used to represent meters and thus "N" was chosen to represent minutes. This is why some of the expressions show a conversion from H (hours) to N (minutes). Notice also that only the first eight characters of the expression-names and location-names are printed. All numbers, integer or floating, are printed here as floating-point.

```
HEAT TRANSFER - SPENT FUEL SHIPPING CASK (WITH MELTABLE LEAD) IN FIRE - FEB 1971

                                ***HEADER***
INITIALZ = YES      LISTINPT = YES      LISTMESH = YES      LISTTEMP = YES
MASS   = POUND      TIME = N=MIN.S      TEMP = 'FAHREN      HEAT     = BTU
LENGTH = INCHES     LISTINCR = 1        SAVETEMP =  NO      SAVEUNIT = 0
MAXITR =  1500      MAXSEC   = 300      NTIMSTEP = 110      STARTIME = 0 S.
STRTSTEP = 1


                                ***MESH***
SYSTEM = R,Z        HORZREGS = 6        VERTREGS = 5
HORZDIST =  4.0 I.   0.5 I.   4.0 I.   0.5 I.   5.0 I.   0.5 I.
HORZNODE =  4        1        4        1        5        1
VERTDIST =  0.5 I.   5.0 I.   0.5 I.   5.0 I.   0.5 I.
VERTNODE =  2        5        2        5        2


                                ***NAMES***
MATERIAL =  1  FOR  SOURCE ,     2  FOR  STEEL ,     3  FOR  LEAD ,
            4  FOR  CONCRETE .


                                ***BOUNDARY***
EXT.TEMP  =  1475.0 'F.  FOR  TOP.SIDE ,     1475.0 'F.  FOR  RHS.SIDE .
GRAD.DIST =     0.6  F.  FOR  TOP.SIDE ,        0.6  F.  FOR  RHS.SIDE .
INSULATED =     0        FOR  BOT.SIDE .


                                ***MATERIALS***
MATRL.NO  =     2        FOR  ALL.MESH ,
                1        FOR  R,Z.REGN  (4,1) ,
                3        FOR  R,Z.REGS  (2,1-3)  (3-4,3) ,
                4        FOR  R,Z.REGS  (2-4,5) .
CONDUCTIVITY  =  125.0 B/FH'F.  FOR  MAT.NOS  1 ,
                  12.0 B/FH'F.  FOR  MAT.NOS  2 ,
                  18.0 B/FH'F.  FOR  MAT.NOS  3 ,
                   0.9 B/FH'F.  FOR  MAT.NOS  4 .
DENSITY       =  169.0 P/FFF.   FOR  MAT.NOS  1 ,
                 500.0 P/FFF.   FOR  MAT.NOS  2 ,
                 710.0 P/FFF.   FOR  MAT.NOS  3 ,
                 180.0 P/FFF.   FOR  MAT.NOS  4 .
SPHEAT        =  0.208 B/P'F.   FOR  MAT.NOS  1 ,
                 0.110 B/P'F.   FOR  MAT.NOS  2 ,
                 0.031 B/P'F.   FOR  MAT.NOS  3 ,
                 0.200 B/P'F.   FOR  MAT.NOS  4 .
MELT.TMP      =  620.0 'F.      FOR  MAT.NOS  3 .
LATNT.HT      =  11.26 B/P.     FOR  MAT.NOS  3 .
CONDUCTIVITY  =    9.2 B/FH'F.  FOR  LIQ.MATL 3 .
DENSITY       =  686.5 P/FFF.   FOR  LIQ.MATL 3 .
SPHEAT        =  0.037 B/P'F.   FOR  LIQ.MATL 3 .
POWER         = 1200.0 W.       FOR  R,Z.REGN  (4,1) .


                                ***TEMPERATURES***
TEMP          =     1   CALL   TO   I/O.FUNCTION  WITH  LUNIT  2 .
AVG.GRP       =     1          FOR  R,Z.NODE  (10,4) ,
                    2          FOR  R,Z.NODE  (8,5)  ,
                    3          FOR  R,Z.NODE  (3,9)  ,
                    4          FOR  R,Z.NODE  (1,20) .


                                ***TIME***
TIM.INCR      =    10 S.        FOR  TIM.STEPS  1-60 ,
                   20 S.        FOR  TIM.STEPS  61-90 ,
                   30 S.        FOR  TIM.STEPS  91-110 .
TEMPFREQ      =    30           FOR  TIM.STEPS  1-60 ,
                   15           FOR  TIM.STEPS  61-90 ,
                   10           FOR  TIM.STEPS  91-110 .


                                ***ENDATA***
```

Figure 4.2

```
                                   **********
                                   * ORTHAT *
*****************************************************************************
*****************************************************************************

 HEAT TRANSFER - SPENT FUEL SHIPPING CASK (WITH MELTABLE LEAD) IN FIRE - FEB 1971

*****************************************************************************
*****************************************************************************

                                        ***HEADER***

INITIALZ        =      YES
LISTINPT        =      YES
LISTMESH        =      YES
LISTTEMP        =      YES
MASS            =      POUND
TIME            =      N=MIN.S
TEMP            =      'FAHREN
HEAT            =      BTU
LENGTH          =      INCHES
LISTINCR        =      1.000000
SAVETEMP        =      NO
SAVEUNIT        =      0.0
MAXITR          =   1500.000000
MAXSEC          =    450.000000
NTIMSTEP        =    110.000000
STARTIME        =      0.0       N.        (     0.0      S.       )
STRTSTEP        =      1.000000


                                        ***MESH***

SYSTEM          =      R,Z
HORZREGS        =      6.000000
VERTREGS        =      5.000000
HORZDIST(  1)   =      4.000000 I.        (     4.00000 I.       )
        (  2)   =      0.500000 I.        (     0.50000 I.       )
        (  3)   =      4.000000 I.        (     4.00000 I.       )
        (  4)   =      0.500000 I.        (     0.50000 I.       )
        (  5)   =      9.000000 I.        (     9.00000 I.       )
        (  6)   =      0.500000 I.        (     0.50000 I.       )
HORZNODE(  1)   =      4.000000
        (  2)   =      1.000000
        (  3)   =      4.000000
        (  4)   =      1.000000
        (  5)   =      9.000000
        (  6)   =      1.000000
VERTDIST(  1)   =      0.500000 I.        (     0.50000 I.       )
        (  2)   =      5.000000 I.        (     5.00000 I.       )
        (  3)   =      0.500000 I.        (     0.50000 I.       )
        (  4)   =      5.000000 I.        (     5.00000 I.       )
        (  5)   =      0.500000 I.        (     0.50000 I.       )
VERTNODE(  1)   =      2.000000
        (  2)   =      5.000000
        (  3)   =      2.000000
        (  4)   =      5.000000
        (  5)   =      2.000000


                                        ***NAMES***

MATERIAL  =        1.00000          (       1.0000          )  FOR  SOURCE
MATERIAL  =        2.00000          (       2.0000          )  FOR  STEEL
MATERIAL  =        3.00000          (       3.0000          )  FOR  LEAD
MATERIAL  =        4.00000          (       4.0000          )  FOR  CONCRETE
```

Figure 4.3

```
                                        ***BOUNDARY***

EXT.TEMP  =     1475.00000 'F.      (   1475.0000 'F.      ) FOR  TOP.SIDE
EXT.TEMP  =     1475.00000 'F.      (   1475.0000 'F.      ) FOR  RHS.SIDE
GRAD.DIS  =        0.60000 F.       (      7.2000 I.       ) FOR  TOP.SIDE
GRAD.DIS  =        0.60000 F.       (      7.2000 I.       ) FOR  RHS.SIDE
INSULATE  =        0.0              (      0.0             ) FOR  BOT.SIDE


                                        ***MATERIALS***

MATRL.NO  =        2.00000          (      2.0000          ) FOR  ALL.MESH
MATRL.NO  =        1.00000          (      1.0000          ) FOR  R,Z.REGN  (4,1)
MATRL.NO  =        3.00000          (      3.0000          ) FOR  R,Z.REGS  (2,1-3) (3-4,3)
MATRL.NO  =        4.00000          (      4.0000          ) FOR  R,Z.REGS  (2-4,5)
CONDUCTI  =      125.00000 B/FH'F.   (      0.1736 B/IN'F.  ) FOR  MAT.NOS   1
CONDUCTI  =       12.00000 B/FH'F.   (      0.0167 B/IN'F.  ) FOR  MAT.NOS   2
CONDUCTI  =       18.00000 B/FH'F.   (      0.0250 B/IN'F.  ) FOR  MAT.NOS   3
CONDUCTI  =        0.90000 B/FH'F.   (      0.0012 B/IN'F.  ) FOR  MAT.NOS   4
DENSITY   =      169.00000 P/FFF.    (      0.0978 P/III.   ) FOR  MAT.NOS   1
DENSITY   =      500.00000 P/FFF.    (      0.2894 P/III.   ) FOR  MAT.NOS   2
DENSITY   =      710.00000 P/FFF.    (      0.4109 P/III.   ) FOR  MAT.NOS   3
DENSITY   =      180.00000 P/FFF.    (      0.1042 P/III.   ) FOR  MAT.NOS   4
SPHEAT    =        0.20800 B/P'F.    (      0.2080 B/P'F.   ) FOR  MAT.NOS   1
SPHEAT    =        0.11000 B/P'F.    (      0.1100 B/P'F.   ) FOR  MAT.NOS   2
SPHEAT    =        0.03100 B/P'F.    (      0.0310 B/P'F.   ) FOR  MAT.NOS   3
SPHEAT    =        0.20000 B/P'F.    (      0.2000 B/P'F.   ) FOR  MAT.NOS   4
MELT.TMP  =      620.00000 'F.       (    620.0000 'F.      ) FOR  MAT.NOS   3
LATNT.HT  =       11.26000 B/P.      (     11.2600 B/P.     ) FOR  MAT.NOS   3
CONDUCTI  =        9.20000 B/FH'F.   (      0.0128 B/IN'F.  ) FOR  LIQ.MATL  3
DENSITY   =      686.50000 P/FFF.    (      0.3973 P/III.   ) FOR  LIQ.MATL  3
SPHEAT    =        0.03700 B/P'F.    (      0.0370 B/P'F.   ) FOR  LIQ.MATL  3
POWER     =     1200.00000 W.        (     68.2464 B.       ) FOR  R,Z.REGN  (4,1)


                                        ***TEMPERATURES***

TEMP      =        1.00000          (      1.0000          ) FOR  I/O.FUNC  2
END  TEMPERATURE  RESTORE  FROM  LUNIT   2 - IMAX,JMAX,KMAX) =    16    20    320
AVG.GRP   =        1.00000          (      1.0000          ) FOR  R,Z.NODE  (10,4)
AVG.GRP   =        2.00000          (      2.0000          ) FOR  R,Z.NODE  (8,5)
AVG.GRP   =        3.00000          (      3.0000          ) FOR  R,Z.NODE  (3,9)
AVG.GRP   =        4.00000          (      4.0000          ) FOR  R,Z.NODE  (1,20)


                                        ***TIME***

TIM.INCR  =       10.00000 S.       (      0.1667 N.       ) FOR  TIM.STEP  1-60
TIM.INCR  =       20.00000 S.       (      0.3333 N.       ) FOR  TIM.STEP  61-90
TIM.INCR  =       30.00000 S.       (      0.5000 N.       ) FOR  TIM.STEP  91-110
TEMPFREQ  =       30.00000          (     30.0000          ) FOR  TIM.STEP  1-60
TEMPFREQ  =       15.00000          (     15.0000          ) FOR  TIM.STEP  61-90
TEMPFREQ  =       10.00000          (     10.0000          ) FOR  TIM.STEP  91-110


                                        ***ENDATA***
```

Fig. 4.3 (continued)

4.3 Printout of Mesh Picture

The next part of the computer printout is a description of the mesh assembly and the materials that make it up. The picture shown in Fig. 4.4 gives a general outline of where the various materials are located within our shipping cask assembly. The letter "H" indicates which regions contain an internal heat source. Although the individual nodes are not shown in this diagram, they are numbered along the top and down the side. A table of the physical properties (conductivity, specific weight, etc.) of each material is printed beneath the picture. The units used in this table correspond to the standard internal units specified earlier. On the next page are printed the region and nodal distances for both the vertical and horizontal directions. The distance from the origin ($r = 0$ or $z = 0$) to the center of each node is given in the right-hand column. The distance from the origin to the end of each region is given in the left-hand column. Of course these distances are measured along the z axis for the vertical direction and along the r axis for the horizontal direction.

MESH ALLOCATION WITH MATERIALS BY REGION

```
REGIONS        1     2     3     4     5     6

               1     5     6    10    11    20
     NODES    TO    TO    TO    TO    TO    TO
               4     5     9    10    19    20
            I-----------------------------------I
        1   I                                   I
  1    TO   I  2     2     2     2     2     2  I
        2   I                                   I
            I---------------     ------         I
        3   I                I     I     I       I
  2    TO   I  3     3     3 I  2  I  4  I  2    I
        7   I                I     I     I       I
            I----------      I     I     I       I
        8   I          I     I     I     I       I
  3    TO   I  2     2 I  3  I  2  I  4  I  2    I
        9   I          I     I     I     I       I
            I-----     I     I     I     I       I
       10   I    I     I     I     I     I       I
  4    TO   I  1 I  2 I  3  I  2  I  4  I  2    I
       14   I  H I     I     I     I     I       I
            I-----     ------      ------        I
       15   I                                   I
  5    TO   I  2     2     2     2     2     2  I
       16   I                                   I
            I-----------------------------------I
```

MATERIAL NO.	MATERIAL NAME	SOLID CONDUCTIVTY B/I'FN.	LIQUID CONDUCTIVTY B/I'FN.	SOLID DENSITY P/III.	LIQUID DENSITY P/III.	SOLID SPEC HEAT B/P'F.	LIQUID SPEC HEAT B/P'F.	MELTNG TEMP 'F.	LATENT HEAT FUSION B/P.
1	SOURCE	0.173611	NOT USED	0.097601	NOT USED	0.208000	NOT USED	NOT USED	NOT USED
2	STEEL	0.016667	NOT USED	0.289352	NOT USED	0.110000	NOT USED	NOT USED	NOT USED
3	LEAD	0.025000	0.012778	0.410880	0.397280	0.031000	0.037000	620.000000	11.260000
4	CONCRETE	0.001250	NOT USED	0.104167	NOT USED	0.200000	NOT USED	NOT USED	NOT USED

Figure 4.4

MESH DISTANCES FOR NODES AND REGIONS (RELATIVE TO ORIGIN)

VERTICAL DIRECTION

VERTICAL REGIONS	VERTICAL NODES	VERTICAL NODAL DISTANCE (I.)
1	1	0.1250
	2	0.3750
0.5000 I.		
2	3	1.0000
	4	2.0000
	5	3.0000
	6	4.0000
	7	5.0000
5.5000 I.		
3	8	5.6250
	9	5.8750
6.0000 I.		
4	10	6.5000
	11	7.5000
	12	8.5000
	13	9.5000
	14	10.5000
11.0000 I.		
5	15	11.1250
	16	11.3750
11.5000 I.		

HORIZONTAL DIRECTION

HORIZONTAL REGIONS	HORIZONTAL NODES	HORIZONTAL NODAL DISTANCE (I.)
1	1	1.0000
	2	2.4142
	3	3.1463
	4	3.7321
4.0000 I.		
2	5	4.2500
4.5000 I.		
3	6	5.1331
	7	6.2835
	8	7.2491
	9	8.0987
8.5000 I.		
4	10	8.7500
9.0000 I.		
5	11	9.6962
	12	11.0056
	13	12.1734
	14	13.2378
	15	14.2223
	16	15.1427
	17	16.0101
	18	16.8327
	19	17.6168
18.0000 I.		
6	20	18.2500
18.5000 I.		

Fig. 4.4 (continued)

4.4 Printout of Iteration Summary

During the execution of both the steady-state and transient codes, a single line may be printed at each iteration summarizing what is happening. Since steady-state iterations involve a search for converged temperatures while transient steps represent the passage of time, their summaries are completely different. Therefore both types of printout will be described for our sample shipping cask. Although there are several pages of such printout, only the first page will be shown here.

The first sixty iterations are shown in Fig. 4.5 for our steady-state problem. The external temperature in this case is 130 °F rather than 1475 °F since the fire has not started yet. As an initial guess we simply assigned every node a starting temperature of 150 °F. The column labeled "MAXIMUM-MESH-TEMPERATURE" shows how quickly this initial guess of 150 °F converges to the final temperature for the one node whose temperature is greatest. The row and column number of this node are given just to the right of the temperature. The first column in this iterative printout is the iteration number, and the second is the cumulative execution time in seconds (32 secs/60 iterations on IBM 360/75; 8 secs/60 iterations on IBM 360/91). The next seven columns tell us how quickly we are moving toward convergence and whether or not we should extrapolate (see Section 2.3). The first column in this group of seven indicates whether we have actually extrapolated (YES or NO). Notice how the "maximum-mesh-temp" suddenly jumps from 232 °F and the "error ratio" drops from .003 to .0006 when the first extrapolation is done after the fiftieth iteration. The "RATIO-SUM" column is just the error term, e_n, given in Eq. (2-19). It should decrease from one iteration to the next. The "ARATIO" column is the extrapolation factor, f, in Eq. (2-20), and must be less than 1.0

ITR	TIME SEC.	TEMP-EXTRAPOLATION-PARAMETERS (RATIO-SUM	ARATIO	ALPHA)	MAXIMUM-ERROR-NODE RATIO	ROW	COL	MAXIMUM-MESH-TEMP (°F)	ROW	COL	WORST REGION HEAT BALANCE (OUTFLOW-GENERATED) B/H	RW	CL	B/H	RW	CL	B/H	RW	CL
1	0	NO	2.73922	0.0	0.0	0.0777906	13	20	155.89246	16	1								
2	1	NO	3.53665	1.29112	-3.4350	0.0897534	13	19	160.53044	16	4								
3	1	NO	3.68474	1.04187	-22.8814	0.0759814	13	18	164.20996	16	4								
4	1	NO	3.60660	0.97879	47.1527	0.0634984	13	17	166.76360	16	4								
5	2	NO	3.35638	0.93062	14.4138	0.0512160	13	16	170.20829	16	1	-2887.0696	4	1	-691.3478	4	3	379.6123	5	1
6	3	NO	3.05991	0.91167	11.3212	0.0398379	13	15	172.56575	16	1								
7	3	NO	2.75596	0.90066	10.0669	0.0295159	13	14	175.06732	16	2								
8	4	NO	2.45855	0.89209	9.2668	0.0242641	13	15	177.38585	16	3								
9	4	NO	2.22508	0.90503	10.5301	0.0233806	12	15	179.84130	16	1								
10	5	NO	2.02073	0.90816	10.8887	0.0232822	12	14	182.19213	16	1	-1651.3693	4	1	-623.2847	4	3	-377.9487	3	3
11	5	NO	1.90098	0.94074	16.8750	0.0234672	12	13	184.35317	16	1								
12	6	NO	1.82859	0.96192	26.2585	0.0222709	12	12	186.20691	16	1								
13	6	NO	1.76092	0.96299	27.0233	0.0190313	11	12	187.86780	16	1								
14	7	NO	1.67815	0.95299	21.2741	0.0146379	7	15	189.58962	16	1								
15	8	NO	1.59306	0.94930	19.7223	0.0157415	7	14	191.38999	16	1	-1428.0498	4	1	-549.0301	4	3	288.1418	4	4
16	8	NO	1.67571	1.05188	-19.2753	0.0167301	7	13	193.03027	16	1								
17	8	NO	1.62235	0.96816	31.4047	0.0172097	10	11	194.45657	16	1								
18	9	NO	1.44959	0.89351	9.3907	0.0111355	6	12	195.96682	16	1								
19	9	NO	1.48035	1.02122	-47.1226	0.0114398	5	12	197.49637	16	1								
20	10	NO	1.41135	0.95339	21.4554	0.0103313	7	11	198.97455	16	1	-1296.7032	4	1	-488.5750	4	3	255.5991	4	4
21	11	NO	1.38254	0.97559	48.9881	0.0094876	6	11	200.38871	16	1								
22	11	NO	1.35515	0.98019	50.4709	0.0079349	7	2	201.80260	16	1								
23	12	NO	1.34256	0.99071	107.6549	0.0077336	4	11	203.20959	16	1								
24	12	NO	1.31131	0.97673	42.9650	0.0084547	3	11	204.57892	16	1								
25	13	NO	1.27253	0.97043	33.8125	0.0074317	5	1	205.89057	16	1	-1182.9550	4	1	-413.6705	4	3	210.5993	4	4
26	13	NO	1.29222	1.01547	-64.6454	0.0069262	8	2	207.18185	16	1								
27	14	NO	1.23813	0.95815	23.8928	0.0069063	7	1	208.51245	16	1								
28	14	NO	1.23580	0.99811	529.6167	0.0066411	7	1	209.75683	16	1								
29	15	NO	1.19819	0.96957	32.8660	0.0063385	7	2	210.98662	16	1								
30	16	NO	1.16354	0.97107	34.5717	0.0062693	8	1	212.20836	16	1	-1061.2040	4	1	-415.8457	4	3	195.5832	4	4
31	16	NO	1.13656	0.97682	43.1335	0.0061688	8	1	213.40364	16	1								
32	17	NO	1.11463	0.98070	51.8176	0.0059837	7	1	214.55793	16	1								
33	17	NO	1.09141	0.97917	48.0135	0.0058086	7	1	215.68742	16	1								
34	17	NO	1.06991	0.98030	50.7629	0.0057208	6	1	216.80046	16	1								
35	18	NO	1.04517	0.97688	43.2497	0.0054699	8	1	217.88454	16	1	-963.1350	4	1	-383.5768	4	3	179.7125	4	4
36	19	NO	1.02357	0.97933	48.3847	0.0054500	7	1	218.95950	16	1								
37	19	NO	1.00281	0.97971	49.2973	0.0051608	8	1	220.00745	16	1								
38	20	NO	0.98151	0.97876	47.0757	0.0050841	7	1	221.03696	16	1								
39	20	NO	0.95868	0.97674	42.9930	0.0049133	8	1	222.04758	16	1								
40	21	NO	0.93706	0.97745	44.3548	0.0047691	8	1	223.03697	16	1	-877.6356	4	1	-351.9003	4	3	163.8924	4	4
41	21	NO	0.91770	0.97934	48.4010	0.0046593	8	1	224.00806	16	1								
42	22	NO	0.89883	0.97943	48.6218	0.0045728	8	1	224.96303	16	1								
43	22	NO	0.87957	0.97857	46.6649	0.0044522	8	1	225.89853	16	1								
44	23	NO	0.86241	0.98049	51.2498	0.0043631	8	1	226.81871	16	1								
45	24	NO	0.84461	0.97937	48.4702	0.0042686	8	1	227.72313	16	1	-801.4250	4	1	-323.2617	4	3	149.8999	4	4
46	24	NO	0.82729	0.97949	48.7608	0.0041674	8	1	228.60993	16	1								
47	25	NO	0.81042	0.97961	49.0346	0.0040737	8	1	229.48070	16	1								
48	25	NO	0.79418	0.97996	49.8933	0.0039852	8	1	230.33647	16	1								
49	25	NO	0.77804	0.97968	49.2018	0.0038976	8	1	231.17664	16	1								
50	26	YES	0.76224	0.97970	49.2657	0.0038143	8	1	232.00188	16	1	-91.4603	4	1	-45.6614	4	3	20.1606	4	4
51	27	NO	0.10011	0.13134	1.1512	0.0006974	4	1	272.01626	14	1								
52	27	NO	0.08971	0.89603	9.6178	0.0004776	15	12	272.11014	14	1								
53	28	NO	0.09440	1.05237	-19.0939	0.0005347	10	12	272.22200	14	1								
54	28	NO	0.08911	0.94390	17.8244	0.0004641	7	11	272.31592	14	1								
55	29	NO	0.08584	0.96329	27.2395	0.0004249	8	1	272.42354	14	1	-92.0941	4	1	-40.6677	4	3	17.5640	4	4
56	29	NO	0.08527	0.99344	152.4157	0.0004083	14	12	272.52078	14	1								
57	30	NO	0.08308	0.97428	38.6775	0.0003866	15	12	272.62140	14	1								
58	30	NO	0.08295	0.99849	660.4085	0.0004640	5	1	272.72007	14	1								
59	31	NO	0.08087	0.97494	39.8984	0.0003867	4	1	272.81696	14	1								
60	32	NO	0.07969	0.98535	68.2391	0.0003737	7	1	272.91288	14	1	-85.2786	4	1	-36.2496	4	3	16.1218	4	4

Figure 4.5

whenever temperatures are to be extrapolated. It is best to let this f factor settle down before extrapolating. The next column, "ALPHA," is simply the f/(1-f) term used in Eq. (2-21). It looks something like our β factor used in Eq. (2-18), although it can take on any value greater than 1.0 whenever extrapolation is allowed. The three columns labeled "MAXIMUM-ERROR-NODE" indicate which node has the greatest error. The "RATIO" value is compared against ϵ as shown in Eq. (2-22) to determine when the temperatures have converged. The last group of columns on the page indicate the three regions which have the worst heat imbalances around their borders. The value printed out is the difference between the heat flowing out of the region and the heat generated within the region. It should be close to zero when convergence is reached. The "RW" and "CL" numbers correspond to the region row and column numbers shown in Fig. 4.4. The frequency of printout is controlled by the HBALINCR parameter explained in Section 3.4 (HBALINCR = 5 here).

The first thirty time steps are shown in Fig. 4.6 for the transient problem. The external temperature has now been increased to 1475 °F to simulate the fire. The column labeled "MAXIMUM OUTSIDE TEMP 'F" is a printout of the maximum outside temperature, T_b or T_{ext}, over the whole boundary (T_{ext} = 1475 °F in this case). The first column in this iterative printout is the step number, and the second is the cumulative machine execution time in seconds (90 secs/30 steps on IBM 360/75; 24 secs/30 steps on IBM 360/91). The third and fourth columns represent the actual fire time being simulated. The third column is always printed in seconds while the fourth is printed in the standard internal units. The next two groups of columns, "MAXIMUM-MESH-TEMPERATURE" and "MINIMUM-MESH-TEMP," are the maximum and minimum nodal temperatures, respectively,

TIME STEP	MACH. TIME (SEC)	REAL TIME (SEC)	REAL TIME (MIN)	MAXIMUM-MESH-TEMPERATURE ('F)	ROW	COL	MATERIAL	MINIMUM-MESH-TEMP ('F)	ROW	COL	MAXIMUM OUTSIDE TEMP 'F	AVERAGE GROUP 1 TEMP 'F	AVERAGE GROUP 2 TEMP 'F	AVERAGE GROUP 3 TEMP 'F	AVERAGE GROUP 4 TEMP 'F
1	3	10.00	0.1667	278.4638	14	1	SOURCE	132.9348	3	19	1475.000	268.661	248.83	197.90	200.16
2	6	20.00	0.3333	278.4639	14	1	SOURCE	134.3105	3	18	1475.000	268.661	248.83	205.18	247.37
3	9	30.00	0.5000	289.2471	1	1	STEEL	135.4609	4	18	1475.000	268.661	248.84	214.26	286.86
4	12	40.00	0.6667	321.7598	1	20	STEEL	135.5808	4	18	1475.000	268.663	248.86	223.83	321.76
5	15	50.00	0.8333	353.5377	1	20	STEEL	135.7731	4	18	1475.000	268.668	248.91	233.45	353.54
6	18	60.00	1.0000	382.9850	1	20	STEEL	136.0512	4	18	1475.000	268.681	249.02	242.93	382.99
7	21	70.00	1.1667	410.5521	1	20	STEEL	136.4269	4	18	1475.000	268.708	249.20	252.22	410.55
8	24	80.00	1.3333	436.5208	1	20	STEEL	136.9102	4	18	1475.000	268.754	249.48	261.28	436.52
9	27	90.00	1.5000	461.0844	1	20	STEEL	137.5096	4	18	1475.000	268.827	249.86	270.10	461.08
10	30	100.00	1.6667	484.3872	1	20	STEEL	138.2322	4	18	1475.000	268.930	250.35	278.68	484.39
11	33	110.00	1.8333	506.5436	1	20	STEEL	139.0784	4	17	1475.000	269.070	250.94	287.04	506.54
12	36	120.00	2.0000	527.6489	1	20	STEEL	139.3924	4	17	1475.000	269.250	251.65	295.17	527.65
13	39	130.00	2.1667	547.7849	1	20	STEEL	139.7577	4	17	1475.000	269.471	252.45	303.09	547.78
14	42	140.00	2.3333	567.0231	1	20	STEEL	140.1771	4	17	1475.000	269.736	253.36	310.80	567.02
15	45	150.00	2.5000	585.4269	1	20	STEEL	140.6528	4	17	1475.000	270.046	254.35	318.31	585.43
16	48	160.00	2.6667	603.0533	1	20	STEEL	141.1870	4	17	1475.000	270.400	255.42	325.62	603.05
17	51	170.00	2.8333	619.9536	1	20	STEEL	141.4336	5	17	1475.000	270.798	256.57	332.76	619.95
18	54	180.00	3.0000	636.1744	1	20	STEEL	141.5811	5	17	1475.000	271.239	257.79	339.71	636.17
19	57	190.00	3.1667	651.7584	1	20	STEEL	141.7494	5	17	1475.000	271.723	259.06	346.50	651.76
20	60	200.00	3.3333	666.7442	1	20	STEEL	141.9400	5	17	1475.000	272.248	260.40	353.12	666.74
21	63	210.00	3.5000	681.1678	1	20	STEEL	142.1541	5	17	1475.000	272.813	261.78	359.58	681.17
22	66	220.00	3.6667	695.0618	1	20	STEEL	142.3928	5	17	1475.000	273.416	263.21	365.90	695.06
23	69	230.00	3.8333	708.4565	1	20	STEEL	142.6573	5	17	1475.000	274.056	264.68	372.06	708.46
24	72	240.00	4.0000	721.3800	1	20	STEEL	142.9488	5	17	1475.000	274.731	266.18	378.09	721.38
25	75	250.00	4.1667	733.8580	1	20	STEEL	143.2682	5	17	1475.000	275.440	267.72	383.98	733.86
26	78	260.00	4.3333	745.9145	1	20	STEEL	143.6165	5	17	1475.000	276.182	269.29	389.75	745.91
27	81	270.00	4.5000	757.5720	1	20	STEEL	143.9946	5	17	1475.000	276.956	270.88	395.39	757.57
28	84	280.00	4.6667	768.8509	1	20	STEEL	144.4034	5	17	1475.000	277.759	272.49	400.90	768.85
29	87	290.00	4.8333	779.7707	1	20	STEEL	144.8438	5	17	1475.000	278.591	274.13	406.30	779.77
30	90	300.00	5.0000	790.3494	1	20	STEEL	145.3163	5	17	1475.000	279.450	275.78	411.59	790.35

Figure 4.6

over the whole mesh at each time step. The row and column numbers of these nodes are printed just to the right of their temperatures. Of course the hottest point is at the outer edge of the steel liner. The last four columns of temperatures, "AVERAGE GROUP 1 'F, etc.," are the temperatures of four corner nodes within the mesh. These four nodes are numbered in Fig. 4.1 from $\odot^{1}$ to $\odot^{4}$. In this way we can follow very closely the temperatures at certain critical spots within the mesh. As was explained in Section 3.9, these "average group temperatures" may be an average of the temperatures over several nodes, as well as individual nodal temperatures. Up to six such temperatures may be printed, with the last two going on another line.

4.5 Printout of Temperature Array

The temperature distributions calculated by ORTHIS and ORTHAT are printed in a maplike framework so that the user can quickly and easily see what his temperatures are anywhere over the whole mesh. The material boundaries are drawn around these temperatures so that one material is separated from another. Whenever the full array is too wide to fit on one page, it is continued on the next page or at the bottom of the current page. Individual nodes or regions may be identified from the row and column numbers printed along the top and down the side. The radial distances are printed just below the last row of temperatures on each page.

Three of the temperature arrays calculated during our transient problem are shown in Fig. 4.7. The first set of temperatures is the initial temperature distribution used at t = 0, the final steady-state temperatures calculated by ORTHIS. The second and third sets are calculated at later times (20 min. and 30 min.) after heat from the fire has started traveling into the center of the shipping cask. Notice how the lead has started to melt along its upper edge. Those nodes at 620 °F are in the process of melting while those greater than 620 °F have already melted.

In some transient-type problems the user is only interested in knowing the general range of temperatures within his mesh. In such cases he may need only the first one or two significant digits for each temperature. By using the LISTTEMP parameter described in Section 3.4, he can print temperature maps instead of the complete temperature distributions. Such a temperature map is shown in Fig. 4.8 for our shipping cask problem after thirty minutes of fire time. After each integer value in this map is

multiplied by the scaling factor of 100, we see that the temperatures range from about 500 °F at the center to around 1300 °F at the outer edges. Another advantage of such a printout is that a large mesh assembly may all fit on one page.

INITIAL TEMPERATURES (°F) AFTER 0 TIME STEPS = 0.0 MIN (0.0 SEC)

REGIONS		1	1	1	1	2	3	3	3	3	4
NODES		1	2	3	4	5	6	7	8	9	10
1	1	203.7881	202.3333	201.3154	200.3543	199.4044	197.6336	194.8157	191.7304	187.3062	179.7835
	2	206.3709	204.8607	203.8042	202.8072	201.8229	199.9911	197.1010	194.0138	189.8207	182.5138
2	3	211.2283	209.6059	208.4709	207.4018	206.3501	204.4039	201.4188	198.4862	195.4421	192.6798
	4	219.4409	217.5534	216.2273	214.9850	213.7797	211.5989	208.5747	206.0570	204.0290	202.6888
	5	228.9883	226.7282	225.0953	223.5495	222.0581	219.4166	216.1267	213.7263	212.0470	211.0146
	6	240.1340	237.4594	235.3796	233.3025	231.2503	227.7006	223.7428	221.1353	219.4637	218.4816
	7	253.1709	250.2176	247.7879	244.8386	241.4960	236.2542	231.1322	228.0490	226.2047	225.1844
3	8	263.5211	261.1306	258.9541	255.6480	248.8296	241.5007	235.3699	231.8888	229.8855	228.8202
	9	269.2724	267.1859	265.3930	262.5787	252.7243	243.5804	237.0054	233.3636	231.2955	230.2114
4	10	273.3161	271.4622	270.0589	268.6607	261.3667	248.5905	240.9183	236.8842	234.6554	233.5236
	11	275.3071	273.6714	272.5075	271.4165	265.6542	253.7494	245.6336	241.2526	238.8312	237.6328
	12	276.8907	275.4086	274.3695	273.3946	268.1118	256.9919	248.8376	244.2490	241.6508	240.3774
	13	277.9582	276.5414	275.5430	274.6045	269.5405	258.8864	250.7255	245.8946	242.9942	241.5384
	14	278.4637	277.0375	275.9986	275.0014	269.9658	259.7581	251.5549	246.3480	242.7476	240.4871
5	15	278.3734	276.9124	275.6979	273.9033	268.8471	259.8874	251.6551	246.2310	241.7460	236.5757
	16	278.3075	276.8136	275.4483	273.2052	268.5940	259.9157	251.6803	246.1906	241.4579	235.8758
HORZDIST(I.)=		1.0000	2.4142	3.1463	3.7321	4.2500	5.1331	6.2835	7.2491	8.0987	8.7500

REGIONS		5	5	5	5	5	5	5	5	5	6
NODES		11	12	13	14	15	16	17	18	19	20
1	1	161.8738	148.7127	141.9979	138.1636	135.7738	134.1832	133.0792	132.3061	131.7933	131.5656
	2	162.6566	149.1771	142.3104	138.3845	135.9339	134.3003	133.1640	132.3655	131.8323	131.6013
2	3	173.6184	156.9311	148.1110	142.7839	139.2615	136.7672	134.9036	133.4466	132.2441	131.6462
	4	187.1730	168.8441	157.7168	150.3458	145.1011	141.1410	137.9935	135.3711	133.0782	131.9461
	5	197.0338	178.1829	165.6508	156.8037	150.2001	145.0205	140.7735	137.1480	133.9355	132.3587
	6	204.9389	185.6759	172.1854	162.2556	154.5954	148.4270	143.2640	138.7903	134.7941	132.8328
	7	211.5392	191.7872	177.5731	166.8355	158.3633	151.4107	145.5007	140.3187	135.6535	133.3559
3	8	214.9687	194.8969	180.3376	169.2308	160.3811	153.0529	146.7732	141.2286	136.2081	133.7250
	9	216.2708	196.0740	181.3877	170.1461	161.1579	153.6906	147.2725	141.5906	136.4339	133.8789
4	10	219.3497	198.8526	183.8775	172.3313	163.0273	155.2395	148.4993	142.4936	137.0113	134.2816
	11	223.1081	202.2854	187.0520	175.2241	165.6063	157.4790	150.3754	143.9773	138.0660	135.0661
	12	225.5627	204.7039	189.4767	177.5884	167.8504	159.5619	152.2615	145.6175	139.3832	136.1779
	13	226.4643	206.0826	191.2094	179.4951	169.8258	161.5594	154.2594	147.5760	141.1853	137.7710
	14	225.2908	206.4408	192.3572	181.0270	171.5777	163.4973	156.4280	150.0491	143.9107	140.2220
5	15	223.1192	206.2658	192.7669	181.6741	172.3599	164.4096	157.5465	151.5914	146.4424	142.6437
	16	223.0660	206.2620	192.7743	181.6860	172.3744	164.4265	157.5674	151.6214	146.5117	143.0879
HORZDIST(I.)=		9.6962	11.0056	12.1734	13.2378	14.2223	15.1427	16.0101	16.8327	17.6168	18.2500

Figure 4.7

NODAL TEMPERATURES (°F) AFTER 90 TIME STEPS = 20.0000 MIN (1200.00 SEC)

REGIONS		1	1	1	1	2	3	3	3	3	4
NODES		1	2	3	4	5	6	7	8	9	10
1	1	697.4589	697.3525	697.2872	697.2742	697.3145	697.4366	698.4701	703.9202	725.2080	785.4683
	2	671.4267	671.3309	671.2728	671.2493	671.2511	671.3157	672.1460	676.7608	695.4102	754.2644
2	3	620.0000	620.0000	620.0000	620.0000	620.0000	620.0000	620.0000	620.0000	620.0000	650.6667
	4	575.4804	575.0316	574.6528	574.2412	573.7620	572.7674	570.7662	568.8025	566.9177	563.5504
	5	529.0496	528.3780	527.7867	527.1028	526.2848	524.4624	520.3587	515.3261	508.9166	499.9981
	6	485.3421	484.6557	484.0848	483.4026	482.5011	480.2172	474.6845	467.9138	459.7953	449.9659
	7	445.1477	444.5808	444.3298	444.2066	443.9636	441.5191	435.2261	427.7365	419.3112	409.9288
3	8	420.2704	419.5659	419.3912	420.1252	423.3556	421.4047	414.8554	407.1762	398.8729	389.9793
	9	407.6428	406.8208	406.5390	407.2502	414.1261	413.8194	407.1392	399.3841	391.1415	382.4538
4	10	399.5794	398.6472	398.1772	397.9288	397.3973	396.2318	388.9560	380.9848	372.9050	364.7313
	11	395.1379	393.7324	392.7390	391.7590	385.8412	376.8378	366.7624	358.0083	350.0500	342.5968
	12	391.0696	389.1199	387.6583	386.1855	377.5392	363.0058	350.1573	340.4301	332.3809	325.4672
	13	387.7624	385.3320	383.4889	381.6535	371.2420	353.0437	337.9842	327.2112	318.7146	312.0024
	14	385.5578	382.7668	380.5659	378.3198	366.3496	346.0825	329.5770	317.7124	307.9261	300.0541
5	15	384.0337	381.1352	378.4851	374.2384	361.5340	343.1166	326.3782	314.0846	302.4894	288.6468
	16	382.9293	379.9343	376.9041	371.5765	360.4182	342.2994	325.5036	313.0797	300.9947	286.5742
HORZDIST(I.)=		1.0000	2.4142	3.1463	3.7321	4.2500	5.1331	6.2835	7.2491	8.0987	8.7500

REGIONS		5	5	5	5	5	5	5	5	5	6
NODES		11	12	13	14	15	16	17	18	19	20
1	1	927.4139	998.5804	1025.6607	1040.5561	1054.4932	1073.1219	1099.7296	1134.3977	1171.0334	1192.0114
	2	913.2445	984.8781	1012.0696	1027.0538	1041.1338	1060.0981	1087.4960	1123.8032	1163.0986	1183.0181
2	3	671.8122	683.4296	691.2733	697.1895	705.8549	725.1751	769.7275	862.0772	1026.0848	1161.4175
	4	434.7975	367.1996	350.5975	345.7999	348.6964	369.7214	436.7518	595.8823	889.4158	1115.9864
	5	335.0407	246.4801	221.8726	212.7629	213.0145	234.2725	309.3141	491.9832	828.3566	1082.4536
	6	294.3208	211.8007	187.0426	176.5858	175.5056	196.0613	272.1650	459.1377	803.2238	1061.8288
	7	275.0201	204.7790	181.8940	171.0289	169.1274	188.8234	264.0794	450.0338	792.4200	1049.5399
3	8	267.8890	204.6748	182.9398	171.9720	169.7191	188.9625	263.5490	448.3110	788.6281	1044.2079
	9	265.3532	204.8236	183.5308	172.5324	170.1548	189.2293	263.5449	447.7980	787.2274	1042.1579
4	10	259.6760	205.5119	185.2735	174.2247	171.5769	190.2638	263.8991	446.8026	783.8636	1037.1013
	11	253.3310	207.0015	188.1381	177.2276	174.5122	192.9102	265.4471	445.4444	777.2480	1027.0930
	12	248.6698	208.4558	191.0950	181.0989	179.4840	198.7876	270.4211	445.2433	766.8419	1010.7249
	13	245.4559	210.9165	196.4660	190.1836	193.8052	218.5358	291.0852	455.0194	751.4512	980.9737
	14	246.2698	219.8910	213.1541	219.0254	239.9309	283.3877	363.9730	504.6301	733.8936	924.2353
5	15	256.3036	237.7922	240.3257	261.6749	303.8613	370.3943	464.2807	585.4118	727.1615	853.2926
	16	256.4422	238.1206	240.8277	262.4623	305.0385	371.9882	466.0984	586.8081	726.3547	840.3870
HORZDIST(I.)=		9.6962	11.0056	12.1734	13.2378	14.2223	15.1427	16.0101	16.8327	17.6168	18.2500

Fig. 4.7 (continued)

FINAL TEMPERATURES (°F) AFTER 110 TIME STEPS = 30.0000 MIN (1800.00 SEC)

REGIONS		1	1	1	1	2	3	3	3	3	4
NODES		1	2	3	4	5	6	7	8	9	10
1	1	811.1912	813.3307	815.1095	817.0167	819.1255	823.4292	832.1562	845.1455	866.3569	902.5922
	2	789.4402	791.5437	793.2969	795.1818	797.2616	801.5205	810.2427	823.6938	843.9380	875.0333
2	3	727.3610	729.0650	730.5004	732.0837	733.8412	737.5325	745.5830	758.3611	771.0996	778.4806
	4	620.0000	620.0000	620.0000	620.0000	620.0000	620.0000	620.9872	659.1519	663.9049	659.7916
	5	590.9373	590.9199	590.9302	590.9900	590.9499	590.9249	591.5478	593.9338	590.8722	581.7626
	6	551.1791	551.0070	550.8581	550.6514	550.3050	549.1880	546.1019	542.0572	535.2712	525.2869
	7	514.3414	514.0773	514.0377	514.1074	514.0550	512.0412	506.6498	499.9610	491.7891	482.0083
3	8	491.3825	490.8131	490.7149	491.4575	494.5032	492.4680	486.1434	478.6212	470.1362	460.6389
	9	479.6994	478.9143	478.6413	479.2980	485.7138	485.0574	478.3616	470.5422	461.9582	452.5791
4	10	472.2158	471.2492	470.7469	470.4611	469.7144	467.8047	459.9694	451.4436	442.6587	433.5764
	11	468.0137	466.5478	465.5097	464.4874	458.3522	448.3739	437.1490	427.2950	418.2024	409.5404
	12	464.1195	462.0845	460.5605	459.0266	450.0420	434.2508	419.7788	408.5207	398.9958	390.6423
	13	460.9389	458.3992	456.4742	454.5569	443.6842	423.9641	406.8771	394.1840	383.8870	375.5708
	14	458.8307	455.9116	453.6102	451.2614	438.7547	416.8258	397.9674	383.7800	371.7001	361.8507
5	15	457.4513	454.4162	451.6403	447.1925	433.9234	413.9708	394.7486	379.9072	365.3691	347.6496
	16	456.4514	453.3132	450.1355	444.5435	432.8079	413.1821	393.8665	378.8288	363.6212	345.0818
HORZDIST(I.)=		1.0000	2.4142	3.1463	3.7321	4.2500	5.1331	6.2835	7.2491	8.0987	8.7500

REGIONS		5	5	5	5	5	5	5	5	5	6
NODES		11	12	13	14	15	16	17	18	19	20
1	1	1018.7639	1084.3040	1112.9229	1131.0802	1148.7041	1170.4929	1198.2048	1230.2542	1260.6335	1276.4386
	2	1006.7312	1072.8714	1101.6973	1120.0464	1137.9591	1160.2862	1188.9755	1222.6092	1255.0962	1270.0816
2	3	794.5964	814.9990	828.1765	839.9394	856.8993	887.7679	943.2644	1033.1523	1162.4312	1254.6385
	4	553.0371	489.5933	472.9645	472.7726	487.8388	531.6542	627.0753	796.7593	1048.1373	1216.6893
	5	427.3538	324.0368	292.1795	285.4899	299.3214	349.7467	465.5463	674.5939	982.7808	1184.5510
	6	364.4159	254.7944	219.4212	210.4620	223.4807	275.9466	398.8380	621.6235	949.1897	1162.1331
	7	331.0465	229.8101	196.2625	187.0377	199.4835	251.9041	375.8747	601.1411	932.2739	1147.3127
3	8	318.3600	224.4716	192.9302	183.9686	196.1851	248.1998	371.5839	596.0026	926.0565	1140.5573
	9	313.8289	222.9185	192.2093	183.3824	195.5271	247.3681	370.4447	594.3730	923.7882	1137.9629
4	10	303.6472	220.1554	191.5634	183.1421	195.1687	246.5700	368.7317	591.1078	918.4784	1131.6183
	11	292.1017	218.9875	193.6830	186.4267	198.9268	249.9416	370.0389	588.1242	909.6933	1120.0744
	12	284.0038	220.1543	198.7720	194.3354	209.3251	261.2660	378.4420	588.2729	897.9518	1102.8138
	13	260.3198	226.9392	212.2301	215.0951	237.8442	294.4669	407.3957	599.6130	881.4057	1073.0730
	14	287.0002	250.3029	248.5350	267.9040	309.1489	378.6624	486.1414	642.1263	857.6915	1018.0426
5	15	305.7059	283.7391	292.8193	327.1303	385.5184	468.1803	574.2692	699.8372	836.3027	951.5704
	16	305.9929	284.3521	293.6358	328.2215	386.9214	469.8170	575.8650	700.8256	835.2803	939.4527
HORZDIST(I.)=		9.6962	11.0056	12.1734	13.2378	14.2223	15.1427	16.0101	16.8327	17.6168	18.2500

Fig. 4.7 (continued)

FINAL TEMPERATURES (°F) AFTER 110 TIME STEPS = 30.0000 MIN (1800.00 SEC)
SCALING FACTOR = 100.00

	REGIONS	1	1	1	1	2	3	3	3	3	4	5	5	5	5	5	5	5	5	5	6
	NODES	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	8	8	8	8	8	8	8	8	9	9	10	11	11	11	11	12	12	12	13	13
	2	8	8	8	8	8	8	8	8	8	9	10	11	11	11	11	12	12	12	13	13
2	3	7	7	7	7	7	7	7	8	8	8	8	8	8	8	9	9	9	10	12	13
	4	6	6	6	6	6	6	6	7	7	7	6	5	5	5	5	5	6	8	10	12
	5	6	6	6	6	6	6	6	6	6	6	4	3	3	3	3	3	5	7	10	12
	6	6	6	6	6	6	5	5	5	5	5	4	3	2	2	2	3	4	6	9	12
	7	5	5	5	5	5	5	5	5	5	5	3	2	2	2	2	3	4	6	9	11
3	8	5	5	5	5	5	5	5	5	5	5	3	2	2	2	2	2	4	6	9	11
	9	5	5	5	5	5	5	5	5	5	5	3	2	2	2	2	2	4	6	9	11
4	10	5	5	5	5	5	5	5	5	4	4	3	2	2	2	2	2	4	6	9	11
	11	5	5	5	5	5	4	4	4	4	4	3	2	2	2	2	2	4	6	9	11
	12	5	5	5	5	5	4	4	4	4	4	3	2	2	2	2	3	4	6	9	11
	13	5	5	5	5	4	4	4	4	4	4	3	2	2	2	2	3	4	6	9	11
	14	5	5	5	5	4	4	4	4	4	4	3	3	2	3	3	4	5	6	9	10
5	15	5	5	5	4	4	4	4	4	4	3	3	3	3	3	4	5	6	7	8	10
	16	5	5	5	4	4	4	4	4	4	3	3	3	3	3	4	5	6	7	8	9

Figure 4.8

4.6 Steady-State Plotting Output

In the steady-state code, ORTHIS, the user may plot temperature maps and profiles using the final converged temperature distribution. The input parameters, MAPPLOT and TRAVPLOT, determine the type of plots to be generated (see Section 3.4). The two plots shown in Figs. 4.9 and 4.10 were generated using the final steady-state temperatures in our shipping cask problem. The first is a map-type plot in which the letters correspond to certain temperature ranges shown in the legend. The region boundaries, rather than material boundaries, are drawn on this plot. The region and node numbers are also given along the top and down the side of the plot. The second figure is a plot of three temperature profiles in the radial direction. Up to five such rows may be plotted on one graph. On the right side of the graph is a legend which indicates which symbols correspond to the three rows, 1, 7, and 14, used in this plot. Row 1 runs along the outer edge of the top steel liner, row 7 travels through the lead just above the inner steel liner, and row 14 passes through the source material at its hottest point. The interfaces between horizontal regions have also been drawn as vertical lines from top to bottom. The individual nodes have been marked off at the top of the graph with tic marks and numbers. Note that the temperatures are plotted at the center of each node. Similar graphs may be obtained for column traverses by using the PLOTCOLS parameter instead of the PLOTROWS parameter.

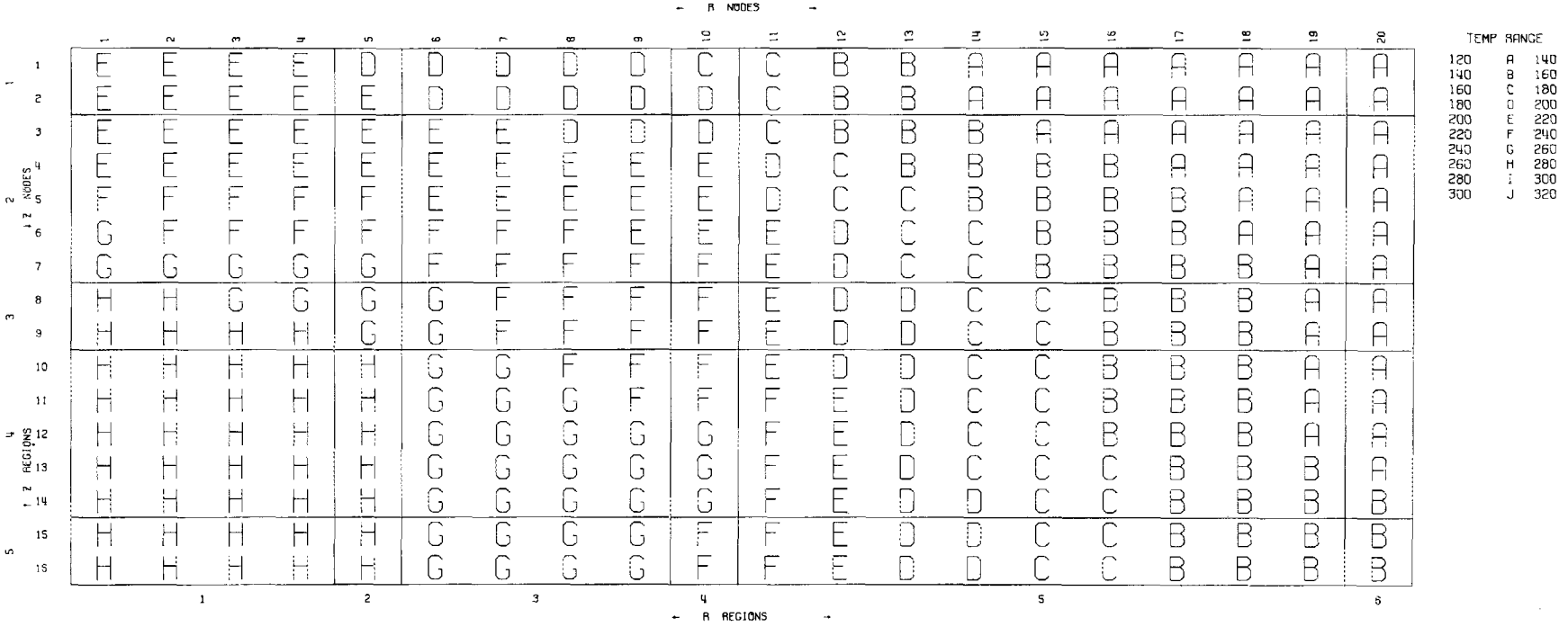

R NODES
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Z NODES
Z REGIONS
1 2 3 4 5 6
R REGIONS
TEMP RANGE
120 A 140
140 B 160
160 C 180
180 D 200
200 E 220
220 F 240
240 G 260
260 H 280
280 I 300
300 J 320

Figure 4.9

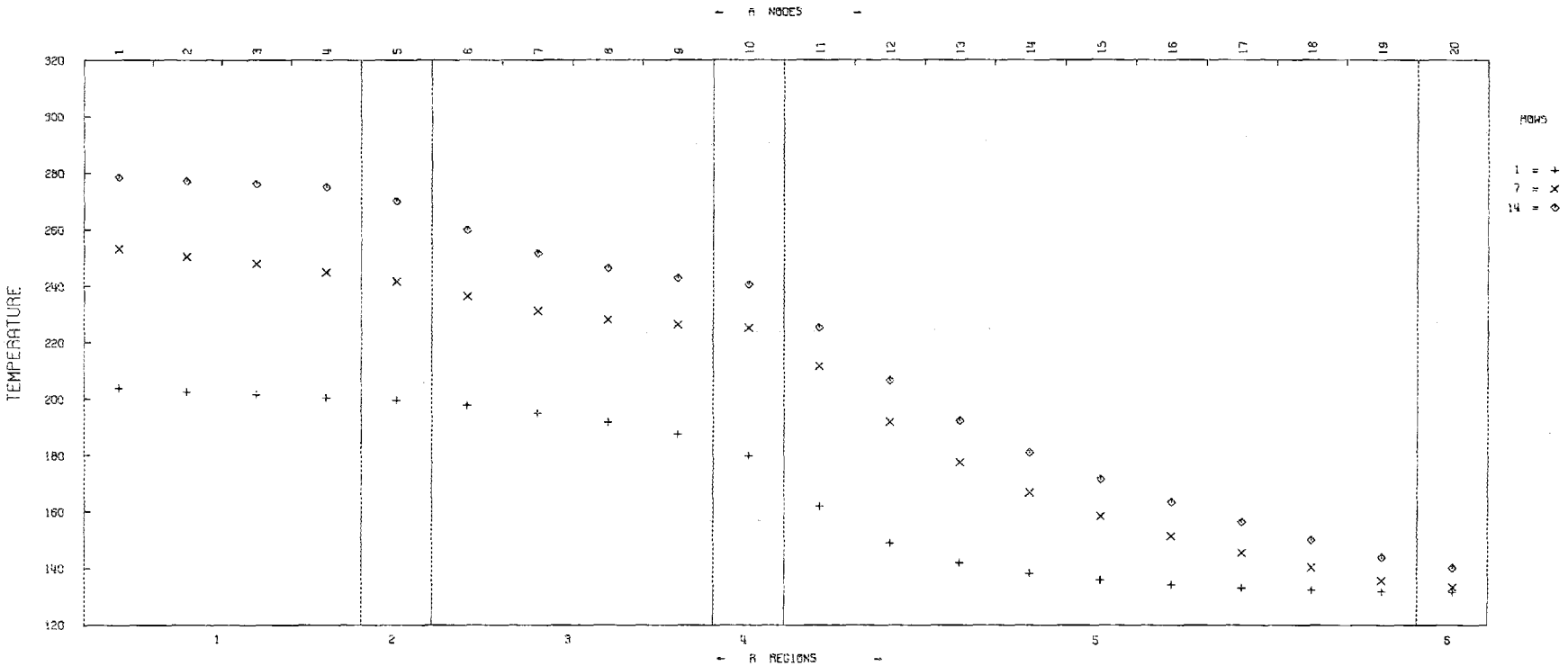

← R NODES →
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
TEMPERATURE
320
300
280
260
240
220
200
180
160
140
120
1 2 3 4 5 6
← R REGIONS →
ROWS
1 = +
7 = ×
14 = ◇

Figure 4.10

5. POSSIBLE FUTURE IMPROVEMENTS

5.1 Dynamic Storage Allocation

There are several areas in which improvements and additions could be made to ORTHIS and ORTHAT. Eight such areas are discussed in Sections 5.1-5.8 of this chapter. Those discussed in the first few sections should probably be done before those in the later sections. Some changes, such as dynamic storage allocation, could be made without much difficulty, while others would require a good deal more work.

To make the best use of our computer facilities and to improve job turnaround, these two codes should first be modified to allocate memory at execution time for the larger FORTRAN arrays. In this way the user could run in the minimum amount of memory necessary to solve his particular problem. In most cases his memory requirements would be much less than the upper limits currently being used (see Section 3.13). Two subroutines, GETCOR and FRECOR,[7] are available to allocate and release core storage within FORTRAN programs during execution. It should not be too difficult to incorporate them into ORTHIS and ORTHAT thus making these two codes also available to other computer installations which have smaller amounts of memory.

5.2 Treatment of Irregular and Internal Boundaries

As was mentioned previously in Section 3.9, the study of cylindrical structures with irregular boundaries or internal boundaries is rather cumbersome when using ORTHIS and ORTHAT. At present, the FIXED TEMP parameter is the best way to treat such problems (see Fig. 3.5). We hope to improve these two codes in the future so that external boundaries with rectangular indentations may be handled more easily. Rectangular cutouts inside the structure itself should also be handled correctly. To do this, radiation across internal boundaries should be considered, as described in the next section. In making these improvements we would plan to keep any additional input as simple and straightforward as possible.

5.3 Treatment of Radiation at Internal and External Boundaries

The transfer of heat from one internal surface to another or from an outside surface to its environment may involve radiation. The normal external boundary condition allowed in ORTHIS and ORTHAT is forced convection, although radiation may be simulated by letting the boundary layer thickness be temperature dependent. This technique is described in Section 2.5. It is also possible to consider radiation across internal boundaries in a similar manner by letting the conductivity be temperature dependent. However, this is again very cumbersome and time-consuming, especially when the number of nodes involved is very large. Therefore it would be very desirable if both codes were able to handle radiation correctly, both across internal rectangular cutouts and across outside boundaries. Since the boundary conditions would then vary with surface temperature rather than remaining fixed, the calculations would take a little longer.

5.4 Automatic Calculation of Time Steps for Transient

In the ***TIME*** input section (see Section 3.10) of the transient code, ORTHAT, the user must specify the length of each time step in seconds, minutes, etc. With this type of procedure, he is not usually able to choose the optimal time steps for his particular problem. Our experience has been that most users chose time steps that are shorter than really necessary. By rerunning experimental problems several times at longer and longer time steps, we found that the temperature distributions usually degraded rather slowly. In one sample problem which involved melting, we found that increasing the time steps by a factor of four resulted in temperature changes of only 0.25%. Therefore, it would be very desirable if the code itself could determine the optimal time steps. Perhaps the user could indicate that he wished to use as large a time step as possible without degrading his temperatures by more than 2%, for example. In this way the code could use longer steps when the temperatures were changing slowly and shorter steps during critical periods when they were changing rapidly. At present the user also specifies the time steps at which temperature distributions are to be printed. By letting the code calculate its own time steps, the user could then ask for temperature distribution printouts at various elapsed times rather than at particular steps. This would eliminate the problem of matching time step iterations with elapsed time.

5.5 Evaluation of Temperature-Dependent Conductivities on Node Interfaces for Transient

In Section 2.5 we discussed the problems which can arise when thermal conductivity is temperature dependent. It was pointed out that Eqs. (2-13) and (2-25) are slightly inaccurate when severe thermal gradients are encountered. This is because the conductivity is assumed to have the same value on all four faces of the node, which is not correct when the temperatures on these interfaces are quite different. Thus the steady-state code was modified to calculate the conductivity on the node interfaces as a function of the interface temperatures, when the conductivity is treated as an external function (CONDF is used).

This modification has not yet been made in the transient code. Since it would cause a significant increase in execution time, it would probably be made available as an option. In temperature-dependent problems where the temperature gradients are not extremely steep, one conductivity value may suffice for all four faces of the node.

5.6 Automatic Extrapolation for Steady-State

In presenting the numerical techniques used in solving the steady-state heat conduction equation (Section 2.3), we described the "dominant-error-mode extrapolation" procedure for accelerating the rate of convergence. This procedure consists of extrapolating the temperatures as shown in Eq. (2-21) using the extrapolation factor, f, in Eq. (2-20). At present, the user must specify how often this extrapolation procedure is to be done. It is usually best to let this f factor settle down before extrapolating. The problem is that the user doesn't know in the beginning how many iterations are required to stabilize this f factor. If the temperatures are extrapolated too frequently, it may result in wild oscillations which slow down the overall rate of convergence. Therefore it would be very desirable if the code itself could examine this f factor and determine when extrapolation of the temperatures would be profitable. Such a modification would not be very difficult, nor would it increase the execution time.

5.7 Complete Regional Heat Rebalancing for Steady-State

In the ***HEADER*** input section (see Section 3.4), an input parameter, HBALINCR, is described for use with the steady-state code, ORTHIS. It controls the frequency at which regional heat imbalances are calculated and printed. These imbalances are calculated for every region in the mesh assembly, by taking the difference between the heat flowing out of each region and the heat generated within each region. These differences should be close to zero when convergence is reached. The three worst regions have their heat imbalances printed out (if HBALINCR $\neq$ 0) as shown in Section 4.4. In the future we may be able to accelerate the rate of convergence, thus driving these imbalances to zero sooner, by adjusting the temperatures within each region accordingly.[3,4] Since most problems have quite a few regions to be considered, we would combine some of them so that the heat rebalancing would be done with a few large regions rather than many small ones.

5.8 Plotting of Temperatures for Transient

In the steady-state code, ORTHIS, two types of temperature plots may be generated. The first is a temperature map of the whole mesh, and the second is a temperature profile of individual rows or columns. In the future we hope to make both of these types available in the transient code so that the spatial distribution of temperatures may be easily seen at specific time steps. In addition, a third type of plot is needed in the transient code to show the variation of temperature with time. Perhaps specific nodes could have their temperatures plotted versus time. The variation of a whole row (or column) of temperatures with time could be shown by taking temperature profiles at different times and plotting them all on the same graph. The first two types of plots could be added to the transient code without much trouble, whereas the third type would require additional programming.

APPENDIX A. NOMENCLATURE

Symbol	Description	English Units
$A_{K,S}$	Area of face S for node K	FF.
a_I	Coefficients along the lower diagonal of the tridiagonal matrix used in transient	B/'F.
$1/a$	Boundary layer thickness	F.
$B_{K,S}$	Internodal distance between node K and the neighbor touching face S	F.
b_I	Coefficients along the main diagonal of the tridiagonal matrix used in transient	B/'F.
β	Relaxation constant ($1.0 \leq \beta < 2.0$) in steady state	
C_K	Specific heat of node K	B/P'F.
c_I	Coefficients along the upper diagonal of the tridiagonal matrix used in transient	B/'F.
e_n	Error term for the nth steady-state iteration	
ϵ	Convergence criterion in steady-state	
F_{se}	Radiation interchange factor	unitless
f	Extrapolation factor in steady-state (e_n/e_{n-1})	
H	Total heat absorbed during the process of melting	B/P.
H_{fu}	Latent heat of fusion	B/P.
H_{LH}	Thickness of each horizontal region	F.
ΔH	Heat increment absorbed at each time step during process of melting	B/P.
h	Heat transfer coefficient at a boundary	B/FFH'F.
h_{LV}	Thickness of each vertical region	F.
I	Vertical node number	

Symbol	Description	English Units
IMAX	Maximum number of vertical nodes	
J	Horizontal node number	
JMAX	Maximum number of horizontal nodes	
K	Internal node number ((I-1)*JMAX + J)	
KMAX	Maximum number of internal nodes (IMAX * JMAX)	
k_K	Thermal conductivity of node K	B/FH'F.
L	Internal region number	
LH	Horizontal region number	
LV	Vertical region number	
m_{LV}	Number of nodes in each vertical region	
n	Iteration number in steady-state	
n_{LH}	Number of nodes in each horizontal region	
$n_{K,S}$	Outward normal of face S for node K	
p_I	Right-hand side of the transient matrix equation	B.
q_K	Heat generation rate per unit volume in node K	B/HFFF.
R_J	Horizontal distance to center of each node	F.
r_J	Horizontal distance to right-hand face of each node	F.
ρ_K	Density of node K	P/FFF.
S	Face number (1,2,3,or 4) for each node	
σ	Stefan-Boltzmann constant (0.174×10^{-8} Btu/hr ft^2 $^\circ$R^4)	B/FFH'R'R'R'R.
T	Temperature	'F.
T_b	Boundary surface temperature	'F.
T_{ext}	External temperature of environment	'F.

Symbol	Description	English Units
T_K	Temperature of node K (at time t in transient)	'F.
$T_K^{-\Delta}$	Temperature of node K at time t - Δt in transient	'F.
T_K^{Δ}	Temperature of node K at time t + Δt in transient	'F.
T_m	Melting temperature	'F.
t	Time	H.
Δt	Time step increment	H.
$\tau_{K,S}$	Temperature of neighboring node touching face S of node K (at time t in transient)	'F.
$\tau_{K,S}^{-\Delta}$	Temperature of neighboring node touching face S of node K at time t - Δt in transient	'F.
$\tau_{K,S}^{\Delta}$	Temperature of neighboring node touching face S of node K at time t + Δt in transient	'F.
Θ_I	Same as Z_I except used in r - θ geometry	radians
θ_I	Same as z_I except used in r - θ geometry	radians
u_K	Heat generated in node K during time interval Δt ($q_K \cdot V_K \cdot \Delta t$)	B.
V_K	Volume of node K	FFF.
v_K	Heat capacity of node K per unit change in temperature ($\rho_K \cdot C_K \cdot V_K$)	B/'F.
$\omega_{K,S}$	Calculation parameter = $k_K \cdot \Delta t \cdot A_{K,S}/2$	BF/'F.
X_J	Same as R_J except used in x-y geometry	F.
x_J	Same as r_J except used in x-y geometry	F.
Y_I	Same as Z_I except used in x-y geometry	F.
y_I	Same as z_I except used in x-y geometry	F.
Z_I	Vertical distance to center of each node	F.
z_I	Vertical distance to bottom face of each node	F.

APPENDIX B. SUMMARY OF INPUT EXPRESSIONS

HEADER

INITIALZ = {YES / NO}	LISTINPT = {YES / NO}	LISTMESH = {YES / NO}
LISTTEMP = {YES / NO}	LISTINCR = {i / 1}	MAXITR = {i / 500}
MAXSEC = {i / 0}	SAVETEMP = {YES / NO}	SAVEUNIT = {i / 0}
LENGTH = {I / F / C / M}	MASS = {P / G / K}	TIME = {S / N / H}
TEMP = {'F / 'R / 'C / 'K}	HEAT = {B / Q / J / E / W}	

Transient Only:

STARTIME = {f [units] / 0.0}	STRTSTEP = {i / 1}	NTIMSTEP = {i / 1}

Steady-State Only:

BETA = {f / 1.5}	EPSILON = {f / .00001}	EXTRINCR = {i / 0}
HBALINCR = {i / 0}	TRAVPLOT = {CRT / MECH / NO}	MAPPLOT = {CRT / MECH / NO}
NTRAVSES = {i / 0}	PLOTCOLS or PLOTROWS = {i i i ... / 0 0 0 ...}	

MESH

SYSTEM = {R,Z / R,T / Y,Z} HORZREGS = {i / 0} VERTREGS = {i / 0}

HORZDIST = {f [units] f [units] f [units] . . . / 0 0 0 . . .}

VERTDIST = {f [units] f [units] f [units] . . . / 0 0 0 . . .}

HORZNODE = {i i i i . . . / 0 0 0 0 . . .}

VERTNODE = {i i i i . . . / 0 0 0 0}

Steady-State Only:

MSWEEP1 = {TOP / BOT} {LEF / RIG} {ROW / COL}

MSWEEP2 = {TOP / BOT} {LEF / RIG} {ROW / COL}

NAMES

MATERIAL NO = {i / 0} FOR {material-name / bbbbbbbb} .

BOUNDARY

[INSULATED / EDGE TEMP / EXT TEMP / GRAD DIST] = {f} [units] FOR (I/O.FUNCTION / {TOP / RHS / BOT / LHS} ÷ {SIDE / REGIONS / NODES}) [Type-1 Integer-expressions] {, / .}

Transient Only:

[TEMP CHANGE / DIST CHANGE] = {f} [units] FOR {I/O.FUNCTION / ALL.TIME / TIM.STEPS} [Type-1 Integer-expressions] {, / .}

MATERIALS

[MATERIAL NO]	= {i}	FOR	I/O.FUNCTION ALL.MESH COL.REGIONS ROW.REGIONS INT.REGIONS	[Type-1 Integer-expression]
			R,Z.REGIONS Y,Z.REGIONS R,T.REGIONS	[Type-2 Integer-expression]
[CONDUCTIVITY *SPHEAT *DENSITY]	= {f, 9999999} [units]	FOR	I/O.FUNCTION ALL.MESH MAT.NOS *LIQ.MATERIALS	[Type-1 Integer-expression]
[*MELTING TEMP *LATENT HEAT]	= {f} [units]	FOR	I/O.FUNCTION ALL.MESH MAT.NOS	[Type-1 Integer-expression]
[HEAT SOURCE POWER LOAD]	= {f, 9999999} [units]	FOR	I/O.FUNCTION ALL.MESH COL / ROW / INT . REGIONS / NODES	[Type-1 Integer-expression]
			R,Z / Y,Z / R,T . REGIONS / NODES	[Type-2 Integer-expression]

*Note: Those keywords with an asterisk are only allowed in the Transient code.

TEMPERATURES

```
[ TEMP          ]
[ FIXED TEMP    ]   = {f} [units] FOR { I/O.FUNCTION                    }   [ Type-1 Integer-  ]  { , }
[ *MAX TEMP     ]                     { ALL.MESH                        }   [ expressions      ]  { . }
                                      { {COL} . {REGIONS}               }
                                      { {ROW}   {NODES}                 }
                                      { {INT}                           }
                                      { {R,Z} . {REGIONS}               }   [ Type-2 Integer-  ]
                                      { {Y,Z}   {NODES}                 }   [ expressions      ]
                                      { {R,T}                           }

[ AVG TEMP GROUP ] = {i}          FOR { I/O.FUNCTION                    }   [ Type-1 Integer-  ]  { , }
                                      { ALL.MESH                        }   [ expressions      ]  { . }
                                      { {COL} . {REGIONS}               }
                                      { {ROW}   {NODES}                 }
                                      { {INT}                           }
                                      { {R,Z} . {REGIONS}               }   [ Type-2 Integer-  ]
                                      { {Y,Z}   {NODES}                 }   [ expressions      ]
                                      { {R,T}                           }
```

*Note: The keyword with an asterisk is only allowed in the Transient code.

TIME

```
[ TIME STEP INCR ] = {f} [units] FOR { I/O.FUNCTION }   [ Type-1 Integer-  ]  { , }
                                     { ALL.TIME     }   [ expression       ]  { . }
                                     { TIM.STEPS    }

[ TEMP LIST FREQ ] = {i}         FOR { I/O.FUNCTION }   [ Type-1 Integer-  ]  { , }
                                     { ALL.TIME     }   [ expression       ]  { . }
                                     { TIM.STEPS    }
```

APPENDIX C. TABLE OF ACCEPTABLE UNITS

Length	Char	Mass	Char	Time	Char
Inches	I	Pound	P	Second	S
Feet	F	Gram	G	Minute	N
Centimeter	C	Kilogram	K	Hour	H
Meter	M				

Temp	Char	Heat	Char
Fahrenheit	'F	BTU	B
Rankine	'R	Calorie	Q
Centigrade	'C	Joule	J
Kelvin	'K	Erg	E
		Watt	W

APPENDIX D

LISTING OF MAIN PROGRAM FOR ORTHIS

```
C     PROGRAM ORTHIS - OAK RIDGE TRANSFER OF HEAT IN STEADY-STATE
C     SPONSOR        - OAK RIDGE NATIONAL LABORATORY
C     PROGRAMMER     - RICHARD C. DURFEE
C     DATE           - JANUARY 1971
C     TITLE          - STEADY STATE SOLUTION OF THE TWO-DIMENSIONAL HEAT
C                      CONDUCTION EQUATION IN R-Z, Y-Z, OR R-THETA GEOMETRY
C     COMPUTER       - IBM/360
C*****BEGIN DESCRIPTION
C     SEE ORNL REPORT
C*****END DESCRIPTION
C
C*****BEGIN VARIABLE DEFINITIONS
C     MHREG  = MAXIMUM NUMBER OF HORIZONTAL REGIONS (IN DIMENSION STATEMENT)
C     MVREG  = MAXIMUM NUMBER OF VERTICAL   REGIONS (IN DIMENSION STATEMENT)
C     MREGS  = MAXIMUM NUMBER OF INTERNAL   REGIONS (IN DIMENSION STATEMENT)
C     MHVREG = MAXIMUM NUMBER OF HORIZ. OR VERT. REGIONS (LARGER OF THE TWO)
C     MHNODE = MAXIMUM NUMBER OF HORIZONTAL NODES   (IN DIMENSION STATEMENT)
C     MVNODE = MAXIMUM NUMBER OF VERTICAL   NODES   (IN DIMENSION STATEMENT)
C     MNODES = MAXIMUM NUMBER OF INTERNAL   NODES   (IN DIMENSION STATEMENT)
C     MHVNOD = MAXIMUM NUMBER OF HORIZ. OR VERT. NODES (LARGER OF THE TWO)
C     MNMATL = MAXIMUM NUMBER OF MATERIALS USED     (IN DIMENSION STATEMENT)
C     MNITR  = MAXIMUM NUMBER OF ITERATIONS         (IN DIMENSION STATEMENT)
C     INITAL = ON-OFF INDICATOR OF WHETHER TO INITIALIZE ALL COMMON AREAS
C     INPRNT = ON-OFF INDICATOR OF WHETHER TO PRINT THE INPUT READ
C     MESHPT = ON-OFF INDICATOR OF WHETHER TO PRINT THE MESH-REGION GRID
C     ITRTMP = ON-OFF INDICATOR OF WHETHER TO PRINT THE ITERATIVE TEMP.S
C     ISAVE  = ON-OFF INDICATOR OF WHETHER TO SAVE THE FINAL TEMPERATURES
C     NITRPR = FREQUENCY OF ITERATIVE PRINTOUT (0 = NO ITERATIVE OUTPUT)
C     LSAVE  = LOGICAL UNIT UPON WHICH FINAL TEMP.S ARE TO BE SAVED
C     LENGTH = INDICATOR 1-4 SHOWING LENGTH UNITS TO BE USED
C     MASS   = INDICATOR 1-3 SHOWING MASS   UNITS TO BE USED
C     ITIME  = INDICATOR 1-3 SHOWING TIME   UNITS TO BE USED
C     ITEMP  = INDICATOR 1-4 SHOWING TEMP.  UNITS TO BE USED
C     IHEAT  = INDICATOR 1-5 SHOWING HEAT   UNITS TO BE USED
C     ITR    = CURRENT ITERATION COUNTER
C     MAXITR = USER SPECIFIED MAXIMUM NUMBER OF ITERATIONS ALLOWED
C     MAXSEC = MAXIMUM NUMBER OF COMPUTER EXECUTION SECONDS ALLOWED
C     NOWTME = CURRENT ELAPSED EXECUTION TIME IN SECONDS
C     INITME = INITIAL STARTUP CLOCK TIME IN SECONDS
C     MAPLOT = INDICATOR FOR PLOTTING TEMP MAP (0=NO, 1=CRT, 2=MECH)
C     IPLTRV = INDICATOR FOR PLOTTING TEMP TRAVERSES (0=NO, 1=CRT, 2=MECH)
C     IAMROW = INDICATOR FOR ROW OR COLUMN TRAVERSES (0=COLUMN, 1=ROW)
C     NROWS  = NUMBER OF ROW (OR COLUMN) TRAVERSES TO BE PLOTTED
C     IROWNO = ROW (OR COLUMN) NODE NUMBERS TO BE PLOTTED (WHOLE ROW PLOTTED)
C     IGEOM  = INDICATOR 1-3 SHOWING GEOMETRY TYPE (1=R,Z 2=Y,Z 3=R,THETA)
C     IMAX   = NUMBER OF HORIZONTAL NODES
C     JMAX   = NUMBER OF VERTICAL   NODES
C     KMAX   = NUMBER OF INTERNAL NODES
C     HRDIST = HORIZONTAL DISTANCES PER,HORIZONTAL REGION
C     VRDIST = VERTICAL   DISTANCES PER,VERTICAL   REGION
C     NHREG  = NUMBER OF HORIZONTAL REGIONS
C     NVREG  = NUMBER OF VERTICAL   REGIONS
C     NTREG  = NUMBER OF INTERNAL   REGIONS
C     NHRNDS = NUMBER OF HORIZONTAL NODES PER REGION
C     NVRNDS = NUMBER OF VERTICAL   NODES PER REGION
C     JTOLH  = HORIZONTAL REGION NUMBER, LH, FROM EACH HORIZ. NODE NO., J
```

```
C      ITOLV  = VERTICAL   REGION NUMBER, LV, FROM EACH VERT.  NODE NO., I
C      LHTOJ  = BEGINNING AND ENDING HORIZ. NODE NO., J, PER HORIZ. REGION
C      LVTOI  = BEGINNING AND ENDING VERT.  NODE NO., I, PER VERT.  REGION
C      RDELTA = THE DIFFERENCE OF THE SQUARES OF THE RADIAL DIST.S PER REGION
C      RO     = R(O)
C      R      = THE RADIAL DISTANCES TO THE NODE INTERSECTIONS
C      RBIGO  = RBIG(O)
C      RBIG   = THE RADIAL DISTANCES TO THE CENTER OF THE NODES
C      ZDELTA = THE VERTICAL NODE DISTANCE PER REGION
C      ZO     = Z(O)
C      Z      = THE VERTICAL DISTANCES TO THE NODE INTERSECTIONS
C      ZBIGO  = ZBIG(O)
C      ZBIG   = THE VERTICAL DISTANCES TO THE CENTER OF THE NODES
C      AREA1  = THE AREA OF FACE 1 OF THE NODES ACROSS ANY HORIZONTAL ROW
C      AREA3  = THE AREA OF FACE 3 OF THE NODES ACROSS ANY HORIZONTAL ROW
C      RAREA4 = THE AREA OF FACE 4 EXCEPT FOR ZDELTA MULTIPLICATION
C      RAREA2 = THE AREA OF FACE 2 EXCEPT FOR ZDELTA MULTIPLICATION
C      VOL    = THE VOLUME OF ANY NODE IN A GIVEN REGION
C      REGVOL = THE TOTAL VOLUME OF EACH REGION
C      TOTVOL = THE TOTAL VOLUME OF THE WHOLE MESH
C      IFT256 = WORK ARRAY FOR DETERMINING NODAL FACE TYPES
C      FTYPE  = FACE TYPE (1-4) OF EACH FACE OF EVERY NODE (CODED 1-255)
C      TOBAR  = EITHER EXT. TEMP. (FOR GRAD.) OR EDGE TEMP. (FOR FIXED TEMP)
C      ARATE  = GRADIENT CONSTANT (1/LENGTH) FOR GRADIENT BOUNDARY CONDITION
C      IBNDRY = INDICATOR SHOWING TYPE OF BOUNDARY CONDITION (FIXED OR GRAD)
C      MTLNAM = EIGHT CHARACTER MATERIAL NAME FOR EACH MATERIAL
C      NMATL  = NUMBER OF MATERIALS
C      MATLNO = MATERIAL NUMBER IN EACH REGION
C      COND   = THERMAL CONDUCTIVITY FOR EACH MATERIAL
C      ICONDF = ON-OFF INDICATOR SHOWING WHETHER COND. IS A FUNCTION ANYWHERE
C      Q      = HEAT SOURCE AT EACH NODE
C      IQREGN = 1-2 INDICATOR SHOWING WHETHER Q IS NON-ZERO IN EACH REGION
C      IQF    = ON-OFF INDICATOR SHOWING WHETHER Q IS A FUNCTION ANYWHERE
C      TMPOLD = TEMPERATURES AT THE PREVIOUS ITERATION
C      TMPNEW = TEMPERATURES CALCULATED FOR THE CURRENT ITERATION
C      TEMP   = TMPNEW FOR THE STEADY STATE CASE
C      AVGTMP = AVERAGE TEMPERATURE FOR ANY OF SIX DIFFERENT GROUPS
C      NAVGRP = NUMBER OF GROUPS TO BE AVERAGED
C      IGRPNO = GROUP NUMBER FOR EACH OF THE SIX GROUPS
C      NAVGND = NUMBER OF NODES TO BE AVERAGED IN EACH GROUP (MAXNOD = 60)
C      MAXNOD = MAXIMUM NUMBER OF NODES THAT CAN BE AVERAGED IN ONE GROUP
C      NODAVG = NODE NUMBERS OF NODES TO BE AVERAGED IN EACH GROUP
C      FIXTMP = TEMPERATURE VALUES TO BE FIXED FOR CERTAIN NODES IN MESH
C      NFIXED = NUMBER OF DIFFERENT TEMPERATURE VALUES TO BE FIXED
C      MAXFIX = MAXIMUM NUMBER OF DIFFERENT TEMP.S TO BE FIXED
C      NODFIX = FIXTMP SUBSCRIPT SHOWING WHICH FIXTMP GOES WITH WHICH NODE
C      IFRST  = INDICATOR SHOWING MESH SWEEP BY ROWS OR COLUMNS
C      IFIRST = INDICATOR SHOWING MESH SWEEP BY ROWS OR COLUMNS (CURRENT)
C      IBEGN  = STARTING VALUE FOR I LOOP (1 OR IMAX)
C      IBEGIN = STARTING VALUE FOR I LOOP (1 OR IMAX) (CURRENT)
C      JBEGN  = STARTING VALUE FOR J LOOP (1 OR JMAX)
C      JBEGIN = STARTING VALUE FOR J LOOP (1 OR JMAX) (CURRENT)
C      INCR   = INCREMENT VALUE FOR I LOOP (1 OR -1)
C      IINCR  = INCREMENT VALUE FOR I LOOP (1 OR -1) (CURRENT)
C      JNCR   = INCREMENT VALUE FOR J LOOP (1 OR -1)
C      JINCR  = INCREMENT VALUE FOR J LOOP (1 OR -1) (CURRENT)
C      ISWTCH = 1-2 SWITCH INDICATING WHICH MESH SWEEP IS CURRENTLY IN USE
C      ICURR  = CURRENT NODAL ROW NUMBER
C      JCURR  = CURRENT NODAL COLUMN NUMBER
C      KCURR  = CURRENT NODAL INTERNAL NODE NUMBER
C      LCURR  = CURRENT NODAL INTERNAL REGION NUMBER
C      IFTYPE = CURRENT NODAL FACE TYPES
C      AREA   = CURRENT NODAL FACE AREAS
C      CONDK  = CURRENT NODAL CONDUCTIVITY
```

```
C      ISIDE  = FACE NO ON WHICH TEMPERATURE-DEPENDENT CONDUCTIVITY IS CALCULATED
C      BS     = CURRENT ADJUSTED INTER-NODAL DISTANCES
C      TMPS   = CURRENT TEMPERATURE OF NEIGHBORING NODES (OR BOUNDARY)
C      QK     = CURRENT NODAL HEAT SOURCE
C      TEMNEW = CURRENT NODAL TEMPERATURE
C      BETA   = RELAXATION CONSTANT (1.5-1.9)
C      ALPHA  = EXTRAPOLATION CONSTANT (.GT. 1.0) = 1/(1-ARATIO)
C      ARATIO = RATIO OF TDIFSM FOR CURRENT AND PREVIOUS ITERATIONS
C      NEXITR = NUMBER OF ITERATIONS BETWEEN EXTRAPOLATIONS
C      IEXTRA = ON-OFF INDICATOR SHOWING WHEN EXTRAPOLATION IS DONE
C      EPSLON = CONVERGENCE CRITERION VALUE COMPARED WITH TERROR
C      TERROR = MAXIMUM ERROR RATIO OVER THE WHOLE MESH
C      TDIFSM = SUM OF ERROR RATIOS OVER THE WHOLE MESH
C      TMPERR = TEMPERATURE AT THE NODE WHERE TERROR IS A MAXIMUM
C      TMPMAX = MAXIMUM TEMPERATURE OVER THE WHOLE MESH
C      IERR   = ROW NUMBER OF THE NODE WHERE TMPERR WAS MEASURED
C      JERR   = COL NUMBER OF THE NODE WHERE TMPERR WAS MEASURED
C      ITPMAX = ROW NUMBER OF THE NODE WHERE TMPMAX WAS MEASURED
C      JTPMAX = COL NUMBER OF THE NODE WHERE TMPMAX WAS MEASURED
C      ICONV  = OFF-ON INDICATOR SHOWING WHEN CONVERGENCE IS REACHED
C      HGEN   = HEAT GENERATED BY INTERNAL SOURCES IN EACH REGION
C      HOUT   = HEAT ESCAPING OUT OF EACH REGION ACROSS THE 4 REGION SIDES
C      HDIFF  = HEAT OUTFLOW - HEAT GENERATED IN THREE WORST REGIONS
C      LVBAD  = VERTICAL REGION NUMBERS OF THREE WORST REGION HEAT BALANCES
C      LHBAD  = HORIZONTAL REGION NUMBERS OF THREE WORST REGION HEAT BALANCES
C      NBALIT = NUMBER OF ITERATIONS BETWEEN HEAT BALANCING (0=NO BALANCING)
C      INTGER = INTEGER WORK ARRAY USED BY INPUT SUBROUTINES
C      KNODES = NODE NUMBERS WORK ARRAY USED BY INPUT SUBROUTINES
C      INTDIM = DIMENSION USED FOR INTGER ARRAY
C      MAXKNO = DIMENSION USED FOR KNODES ARRAY
C      EXPSET = SET OF 'FOR-EXPRESSION' EXPRESSION NAMES
C      LOCSET = SET OF 'FOR-EXPRESSION' LOCATION   NAMES
C      NEXP   = NUMBER OF EXPRESSION NAMES
C      NLOC   = NUMBER OF LOCATION   NAMES
C      VARNMS = SET OF 'ARITHMETIC-EXPRESSION' VARIABLE NAMES
C      NVAR   = NUMBER OF VARIABLE NAMES
C      NUMERC = ON-OFF INDICATOR SHOWING WHEN VARNMS IS NUMERIC OR ALPHAMERIC
C      NVAL   = NUMBER OF VALUES TO RHS OF EQUAL SIGN FOR EACH VARNMS
C      VARVAL = RHS VALUES OF EACH 'ARITH.-EXPRESSION'
C      TITLE  = ALPHAMERIC TITLE CARDS READ IN FOR EACH CASE
C*****END VARIABLE DEFINITIONS
C
      IMPLICIT REAL*8 (A-H,O-Z)
C*****BEGIN MAXIMUM DIMENSIONS COMMON AREA
      COMMON /GEOM/   MHREG ,MVREG ,MREGS ,MHVREG,
     1                MHNODE,MVNODE,MNODES,MHVNOD
      COMMON /MATERL/ MNMATL
      COMMON /ITERAT/ MNITR
C*****END MAXIMUM DIMENSIONS COMMON AREA
C
C*****BEGIN PARAMETER COMMON AREA
      COMMON /RERUN/  INITAL
      COMMON /PRINT/  INPRNT,MESHPT,NITRPR,ITRTMP
      COMMON /SAVE/   ISAVE ,LSAVE
      COMMON /UNITS/  LENGTH,MASS  ,ITIME ,ITEMP  ,IHEAT
      COMMON /ITRATE/ ITR,MAXITR
      COMMON /EXECTM/ MAXSEC,NOWTME,INITME
      COMMON /PLOTMP/ MAPLOT,IPLTRV
      COMMON /TRVRSE/ IAMROW,NROWS,IROWNO(5)
C*****END PARAMETER COMMON AREA
C
C*****BEGIN MESH GEOMETRY COMMON AREA
      COMMON /COORD/  IGEOM
      COMMON /IJKMAX/ IMAX,JMAX,KMAX
```

```
      COMMON /REGION/ HRDIST(50),VRDIST(50),
     1                NHREG     ,NVREG     ,NTREG,
     2                NHRNDS(50),NVRNDS(50),
     3                JTOLH(1000),ITOLV(1000),
     4                LHTOJ(50,2),LVTOI(50,2)
      INTEGER*2       ITOLV,JTOLH,LVTOI,LHTOJ
      COMMON /MESH/   RDELTA(50),R0,R(1000),RBIG0,RBIG(1000),
     1                ZDELTA(50),Z0,Z(1000),ZBIG0,ZBIG(1000)
      COMMON /FAREAS/ AREA1(1000),RAREA4(1001)
      DIMENSION       AREA3(1000),RAREA2(1000)
      EQUIVALENCE    (AREA1, AREA3),(RAREA4(2),RAREA2(1))
C     AREA4 = RAREA4*ZDELTA
C     AREA2 = RAREA2*ZDELTA
      COMMON /NDVOLS/ VOL(2500),REGVOL(2500),TOTVOL
      COMMON /FACES/  IFT256(4,256),FTYPE(10000)
      LOGICAL*1       FTYPE
      INTEGER*2       IFT256
C*****END MESH GEOMETRY COMMON AREA
C
C*****BEGIN BOUNDARY CONDITIONS COMMON AREA
      COMMON /BNDARY/ TOBAR(4,1000),ARATE(4,1000),IBNDRY(4,1000)
      LOGICAL*1       IBNDRY
C*****END BOUNDARY CONDITIONS COMMON AREA
C
C*****BEGIN MATERIAL PROPERTIES COMMON AREA
      COMMON /REGMTL/ MTLNAM(300),NMATL,MATLNO(2500)
      REAL*8          MTLNAM
      COMMON /MTCOND/ COND(300),ICONDF
      COMMON /HEATQ/  Q(10000),IQREGN(2500),IQF
C*****END MATERIAL PROPERTIES COMMON AREA
C
C*****BEGIN TEMPERATURE COMMON AREA
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)
      DIMENSION       TEMP  (10000)
      EQUIVALENCE    (TMPNEW,TEMP)
C     TEMP = BEST (NEWEST) TEMPERATURES FOR STEADY STATE                STEADY
      COMMON /AVGTEM/ AVGTMP(6),NAVGRP,IGRPNO(6),NAVGND(6),MAXNOD,
     1                NODAVG(6,60)
C     MAXNOD = 60 = MAXIMUM NUMBER OF NODES AVERAGED IN ONE GROUP
      COMMON /FIXTEM/ FIXTMP(255),NFIXED,MAXFIX,NODFIX(10000)
      LOGICAL*1 NODFIX
C     MAXFIX = 255 = MAXIMUM NO. OF DIFFERENT TEMP.S TO BE FIXED (NODFIX=BYTE)
C*****END TEMPERATURE COMMON AREA
C
C*****BEGIN MESH SWEEP COMMON AREA
      COMMON /MSWEEP/ IFRST(2),IBEGN(2),INCR(2),JBEGN(2),JNCR(2),
     1                IFIRST  ,IBEGIN  ,IINCR  ,JBEGIN  ,JINCR  ,ISWTCH
C*****END MESH SWEEP COMMON AREA
C
C*****BEGIN NODAL INFORMATION COMMON AREA
      COMMON /NDIJKL/ ICURR,JCURR,KCURR,LCURR
      COMMON /NDFACE/ IFTYPE(4)
      COMMON /NDAREA/ AREA(4)
      COMMON /NDCOND/ CONDK,ISIDE
      COMMON /NDBS/   BS(4)
      COMMON /NDTMPS/ TMPS(4)
      COMMON /NDQ/    QK
      COMMON /TEMPK/  TEMNEW
C*****END NODAL INFORMATION COMMON AREA
C
C*****BEGIN RELAXATION-EXTRAPOLATION-CONVERGENCE COMMON AREA
      COMMON /RELAX/  BETA
      COMMON /EXTRAP/ ALPHA,ARATIO,NEXITR,IEXTRA
      COMMON /CONVER/ EPSLON,TERROR(1500),TDIFSM(1500),TMPERR,TMPMAX,
```

```
     1                IERR,JERR,ITPMAX,JTPMAX,ICONV
C*****END RELAXATION-EXTRAPOLATION-CONVERGENCE COMMON AREA
C
C*****BEGIN REGION HEAT BALANCING COMMON AREA
      COMMON /REGBAL/ HGEN(2500),HOUT(2500),HDIFF(3),LVBAD(3),LHBAD(3)
      COMMON /BALITR/ NBALIT
C*****END REGION HEAT BALANCING COMMON AREA
C
C*****BEGIN WORK ARRAYS COMMON AREA
      COMMON /MAXDIM/ INTDIM,MAXKNO
C     INTDIM =  500 = COLUMN DIMENSION OF THE ARRAY, INTGER
C     MAXKNO = 5000 = DIMENSION OF THE ARRAY, KNODES
      COMMON /WORK1/  INTGER(2,500)
      COMMON /WORK2/  KNODES(5000)
      INTEGER*2       INTGER,KNODES
      COMMON /EXPRES/ EXPSET(12),NEXP
      COMMON /LOCATE/ LOCSET(28),NLOC
      REAL*8          LOCSET
      COMMON /AEXPNM/ VARNMS(33),NVAR,NUMERC(33),NVAL(33)
      COMMON /HVDIST/ VARVAL(50)
      COMMON /HEADNG/ TITLE(10,12)
C*****END WORK ARRAYS COMMON AREA
C
C*****BEGIN INPUT
      CALL INITLZ
 1    CALL INPUT
      IF(MESHPT .EQ. 1) CALL MESHPR
C*****END INPUT
C*****BEGIN INITIALIZATION
      INITME=ICLOCK(0)/100
C     CALL SPACNG  IS DONE INSIDE OF SUBROUTINE INPUT
C     CALL AREAS   IS DONE INSIDE OF SUBROUTINE INPUT
C     CALL VOLUME  IS DONE INSIDE OF SUBROUTINE INPUT
      CALL NDTYPE
      ICONV=0
      ISWTCH=0
      ITR=0
C*****END INITIALIZATION
C
C*****BEGIN OUTER TEMPERATURE CONVERGENCE ITERATION LOOP ***********************
 2    ITR=ITR+1
C*****BEGIN SETUP FOR THE TYPE OF MESH SWEEP
      ISWTCH=ISWTCH+1
      IF(ISWTCH .GT. 2) ISWTCH=1
      IFIRST=IFRST(ISWTCH)
      IBEGIN=IBEGN(ISWTCH)
      IINCR =INCR (ISWTCH)
      JBEGIN=JBEGN(ISWTCH)
      JINCR =JNCR (ISWTCH)
      IF(IFIRST .EQ. 1) IJMAX=IMAX
      IF(IFIRST .NE. 1) IJMAX=JMAX
      IF(IFIRST .EQ. 1) JIMAX=JMAX
      IF(IFIRST .NE. 1) JIMAX=IMAX
C*****END SETUP FOR THE TYPE OF MESH SWEEP
C
C*****BEGIN MESH SWEEP
      IF(IFIRST .EQ. 1) I=IBEGIN-IINCR
      IF(IFIRST .NE. 1) J=JBEGIN-JINCR
      DO 11 IJ=1,IJMAX
      IF(IFIRST .NE. 1) GO TO 3
      I=I+IINCR
      J=JBEGIN-JINCR
      GO TO 4
 3    J=J+JINCR
```

```
      I=IBEGIN-IINCR
 4    DO 10 JI=1,JIMAX
      IF(IFIRST .NE. 1) GO TO 5
      J=J+JINCR
      GO TO 6
 5    I=I+IINCR
C*****BEGIN NODE DESCRIPTION CALCULATIONS
 6    K=(I-1)*JMAX+J
      LV=ITOLV(I)
      LH=JTOLH(J)
      L=(LV-1)*NHREG+LH
      MATL=MATLNO(L)
      ICURR=I
      JCURR=J
      KCURR=K
      LCURR=L
C     I,J,K    = NODE   POSITION
C     LV,LH,L  = REGION POSITION
C     MATL     = NODE   MATERIAL
C*****END NODE DESCRIPTION CALCULATIONS
C*****BEGIN FIXED TEMP. NODES CHECK
      IFIX=NODFIX(K)
      IF(IFIX .EQ. 0) GO TO 7
      TMPNEW(K)=FIXTMP(IFIX)
      TEMNEW   =TMPNEW(K)
      GO TO 10
C*****END FIXED TEMP. NODES CHECK
C*****BEGIN FIRST TEMPERATURE CALCULATIONS
 7    ISIDE=0
      IF(ICONDF .EQ. 0) CONDK=GTCOND(MATL)
      QK   =GETQK (K)
      CALL  GETYPE(K)
      CALL  GTAREA(I,J)
      CALL  GETBS (I,J)
      CALL  GETMPS(I,J,K)
      TMPNEW(K)=GETEMP(L)
      TEMNEW   =TMPNEW(K)
C*****END FIRST TEMPERATURE CALCULATIONS
      IF(ICONDF .NE. 0) GO TO 8
      IF(IQF    .NE. 0) GO TO 8
      GO TO 9
C*****BEGIN SECOND TEMPERATURE CALCULATIONS - BETTER CONDUCTIVITY, HEAT SOURCE
 8    IF(IQF .NE. 0) QK=GETQK(K)
      CALL  GETBS(I,J)
      TMPNEW(K)=GETEMP(L)
      TEMNEW   =TMPNEW(K)
C*****END SECOND TEMPERATURE CALCULATIONS - BETTER CONDUCTIVITY, HEAT SOURCE
C*****BEGIN BETA RELAXATION APPLICATION
 9    TMPNEW(K) = BETA*TMPNEW(K) + (1.0D+00 - BETA)*TMPOLD(K)
      TEMNEW    = TMPNEW(K)
C*****END BETA RELAXATION APPLICATION
 10   CONTINUE
 11   CONTINUE
C*****END MESH SWEEP
C
C*****BEGIN CONVERGENCE TEST
      CALL CONVRG
C*****BEGIN ALPHA EXTRAPOLATION
      IEXTRA=0
      IF(ITR .EQ. 1) GO TO 14
      INEW=0
      IF(NEXITR .EQ. 0) GO TO 12
      IDIV=ITR/NEXITR
      INEW=IDIV*NEXITR
```

```
 12   ARATIO=TDIFSM(ITR)/TDIFSM(ITR-1)
      ALPHA=0.0D+00
      IF(ARATIO .NE. 1.0D+00) ALPHA=1.0D+00/(1.0D+00-ARATIO)
      IF(INEW .NE. ITR) GO TO 14
      IF(ARATIO .GE. 1.0D+00) GO TO 14
      DO 13 K=1,KMAX
      IFIX=NODFIX(K)
      IF(IFIX .NE. 0) GO TO 13
      TMPNEW(K) = ALPHA*TMPNEW(K) + (1.0D+00 - ALPHA)*TMPOLD(K)
 13   CONTINUE
      IEXTRA=1
 14   CONTINUE
C*****END ALPHA EXTRAPOLATION
C*****BEGIN REGION HEAT BALANCING
      DO 15 I=1,3
      HDIFF(I)=0.0D+00
      LVBAD(I)=0
 15   LHBAD(I)=0
      IF(NBALIT .EQ. 0) GO TO 16
      IDIV=ITR/NBALIT
      INEW=IDIV*NBALIT
      IF(INEW .EQ. ITR) CALL BALANC
 16   CONTINUE
C*****END REGION HEAT BALANCING
C*****BEGIN ITERATION OUTPUT
      NOWTME=ICLOCK(0)/100-INITME
      CALL OUTPUT(0)
C*****END ITERATION OUTPUT
      IF(ICONV .EQ. 1) GO TO 20
C*****END CONVERGENCE TEST
C*****BEGIN EXECUTION TIME TEST AND MAXIMUM ITERATIONS TEST
      IF(MAXSEC .EQ. 0) GO TO 17
      IF(NOWTME .LT. MAXSEC) GO TO 17
      PRINT 1001,NOWTME,MAXSEC,ITR
 1001 FORMAT(//1X,'EXECUTION TIME =',I6,' SEC.S HAS EXCEEDED MAXIMUM ALL
     1OWABLE TIME =',I6,' SEC.S AFTER ITERATION =',I6//)
      GO TO 20
 17   IF(ITR .LT. MAXITR) GO TO 18
      PRINT 1002,ITR,MAXITR,ITR
 1002 FORMAT(//1X,'NUMBER OF ITERATIONS =',I6,' HAS EXCEEDED MAXIMUM ALL
     1OWABLE ITERATIONS =',I6,' AFTER ITERATION =',I6//)
      GO TO 20
C*****END EXECUTION TIME TEST AND MAXIMUM ITERATIONS TEST
C*****BEGIN RESETTING TMPOLD
 18   DO 19 K=1,KMAX
 19   TMPOLD(K)=TMPNEW(K)
C*****END RESETTING TMPOLD
      GO TO 2
C*****END OUTER TEMPERATURE CONVERGENCE ITERATION LOOP ************************
C
C*****BEGIN FINAL OUTPUT
 20   CALL OUTPUT(1)
C*****END FINAL OUTPUT
C*****BEGIN FINAL SAVE OUTPUT
      IF(ISAVE .EQ. 0) GO TO 21
      CALL TPSAVE
C*****END FINAL SAVE OUTPUT
 21   PRINT 1003
 1003 FORMAT(1H1)
      GO TO 1
      END
```

LISTING OF MAIN PROGRAM FOR ORTHAT

```
C     PROGRAM ORTHAT - OAK RIDGE TRANSFER OF HEAT AT TRANSIENT
C     SPONSOR     - OAK RIDGE NATIONAL LABORATORY
C     PROGRAMMER - RICHARD C. DURFEE
C     DATE        - JANUARY 1971
C     TITLE       - TRANSIENT SOLUTION OF THE TWO-DIMENSIONAL HEAT
C                   CONDUCTION EQUATION IN R-Z, Y-Z, OR R-THETA GEOMETRY
C                   BY A PREDICTOR-CORRECTOR MODIFICATION TO THE
C                   ALTERNATING DIRECTION METHOD
C     COMPUTER    - IBM/360
C*****BEGIN DESCRIPTION
C     SEE ORNL REPORT
C*****END DESCRIPTION
C
C*****BEGIN VARIABLE DEFINITIONS
C     MHREG  = MAXIMUM NUMBER OF HORIZONTAL REGIONS (IN DIMENSION STATEMENT)
C     MVREG  = MAXIMUM NUMBER OF VERTICAL   REGIONS (IN DIMENSION STATEMENT)
C     MREGS  = MAXIMUM NUMBER OF INTERNAL   REGIONS (IN DIMENSION STATEMENT)
C     MHVREG = MAXIMUM NUMBER OF HORIZ. OR VERT. REGIONS (LARGER OF THE TWO)
C     MHNODE = MAXIMUM NUMBER OF HORIZONTAL NODES   (IN DIMENSION STATEMENT)
C     MVNODE = MAXIMUM NUMBER OF VERTICAL   NODES   (IN DIMENSION STATEMENT)
C     MNODES = MAXIMUM NUMBER OF INTERNAL   NODES   (IN DIMENSION STATEMENT)
C     MHVNOD = MAXIMUM NUMBER OF HORIZ. OR VERT. NODES (LARGER OF THE TWO)
C     MNMATL = MAXIMUM NUMBER OF MATERIALS USED     (IN DIMENSION STATEMENT)
C     MNITR  = MAXIMUM NUMBER OF ITERATIONS         (IN DIMENSION STATEMENT)
C     INITAL = ON-OFF INDICATOR OF WHETHER TO INITIALIZE ALL COMMON AREAS
C     INPRNT = ON-OFF INDICATOR OF WHETHER TO PRINT THE INPUT READ
C     MESHPT = ON-OFF INDICATOR OF WHETHER TO PRINT THE MESH-REGION GRID
C     NITRPR = FREQUENCY OF ITERATIVE PRINTOUT (0 = NO ITERATIVE OUTPUT)
C     ITRTMP = ON-OFF INDICATOR OF WHETHER TO PRINT TEMP. ARRAY OR TEMP. MAP
C     ISAVE  = ON-OFF INDICATOR OF WHETHER TO SAVE THE FINAL TEMPERATURES
C     LSAVE  = LOGICAL UNIT UPON WHICH FINAL TEMP.S ARE TO BE SAVED
C     LENGTH = INDICATOR 1-4 SHOWING LENGTH UNITS TO BE USED
C     MASS   = INDICATOR 1-3 SHOWING MASS   UNITS TO BE USED
C     ITIME  = INDICATOR 1-3 SHOWING TIME   UNITS TO BE USED
C     ITEMP  = INDICATOR 1-4 SHOWING TEMP.  UNITS TO BE USED
C     IHEAT  = INDICATOR 1-5 SHOWING HEAT   UNITS TO BE USED
C     ITR    = CURRENT ITERATION COUNTER FOR TIME STEPS
C     MAXITR = USER SPECIFIED MAXIMUM NUMBER OF ITERATIONS ALLOWED
C     MAXSEC = MAXIMUM NUMBER OF COMPUTER EXECUTION SECONDS ALLOWED
C     NOWTME = CURRENT ELAPSED EXECUTION TIME IN SECONDS
C     INITME = INITIAL STARTUP CLOCK TIME IN SECONDS
C     IGEOM = INDICATOR 1-3 SHOWING GEOMETRY TYPE (1=R,Z 2=Y,Z 3=R,THETA)
C     IMAX   = NUMBER OF HORIZONTAL NODES
C     JMAX   = NUMBER OF VERTICAL   NODES
C     KMAX   = NUMBER OF INTERNAL NODES
C     HRDIST = HORIZONTAL DISTANCES PER,HORIZONTAL REGION
C     VRDIST = VERTICAL   DISTANCES PER,VERTICAL   REGION
C     NHREG  = NUMBER OF HORIZONTAL REGIONS
C     NVREG  = NUMBER OF VERTICAL   REGIONS
C     NTREG  = NUMBER OF INTERNAL   REGIONS
C     NHRNDS = NUMBER OF HORIZONTAL NODES PER REGION
C     NVRNDS = NUMBER OF VERTICAL   NODES PER REGION
C     JTOLH  = HORIZONTAL REGION NUMBER, LH, FROM EACH HORIZ. NODE NO., J
C     ITOLV  = VERTICAL   REGION NUMBER, LV, FROM EACH VERT.  NODE NO., I
C     LHTOJ  = BEGINNING AND ENDING HORIZ. NODE NO., J, PER HORIZ. REGION
C     LVTOI  = BEGINNING AND ENDING VERT.  NODE NO., I, PER VERT.  REGION
C     RDELTA = THE DIFFERENCE OF THE SQUARES OF THE RADIAL DIST.S PER REGION
C     RO     = R(O)
C     R      = THE RADIAL DISTANCES TO THE NODE INTERSECTIONS
C     RBIGO  = RBIG(O)
C     RBIG   = THE RADIAL DISTANCES TO THE CENTER OF THE NODES
C     ZDELTA = THE VERTICAL NODE DISTANCE PER REGION
```

```
C     ZO      = Z(C)                                                    630
C     Z       = THE VERTICAL DISTANCES TO THE NODE INTERSECTIONS        640
C     ZBIGO   = ZBIG(C)                                                 650
C     ZBIG    = THE VERTICAL DISTANCES TO THE CENTER OF THE NODES       660
C     AREA1   = THE AREA OF FACE 1 OF THE NODES ACROSS ANY HORIZONTAL ROW 670
C     AREA3   = THE AREA OF FACE 3 OF THE NODES ACROSS ANY HORIZONTAL ROW 680
C     RAREA4  = THE AREA OF FACE 4 EXCEPT FOR ZDELTA MULTIPLICATION     690
C     RAREA2  = THE AREA OF FACE 2 EXCEPT FOR ZDELTA MULTIPLICATION     700
C     VOL     = THE VOLUME OF ANY NODE IN A GIVEN REGION                710
C     REGVOL  = THE TOTAL VOLUME OF EACH REGION                         720
C     TOTVOL  = THE TOTAL VOLUME OF THE WHOLE MESH                      730
C     IFT256  = WORK ARRAY FOR DETERMINING NODAL FACE TYPES             740
C     FTYPE   = FACE TYPE (1-4) OF EACH FACE OF EVERY NODE (CODED 1-255) 750
C     TOBAR   = EITHER EXT. TEMP. (FOR GRAD.) OR EDGE TEMP. (FOR FIXED TEMP) 760
C     ARATE   = GRADIENT CONSTANT (1/LENGTH) FOR GRADIENT BOUNDARY CONDITION 770
C     IBNDRY  = INDICATOR SHOWING TYPE OF BOUNDARY CONDITION (FIXED OR GRAD) 780
C     TEMPCH  = BOUNDARY TEMP. CHANGE CONSTANT SENT TO SUBR. TOBARF FOR EACH STEP
C     DISTCH  = BOUNDARY GRAD. CHANGE CONSTANT SENT TO SUBR. ARATEF FOR EACH STEP
C     TINITL  = INITIAL BOUNDARY TEMP.S SET BY EXT.TEMP OR EDGE.TEMP INPUT CARDS
C     MTLNAM  = EIGHT CHARACTER MATERIAL NAME FOR EACH MATERIAL         820
C     NMATL   = NUMBER OF MATERIALS                                     830
C     MATLNO  = MATERIAL NUMBER IN EACH REGION                          840
C     COND    = THERMAL CONDUCTIVITY FOR EACH MATERIAL BY STATE (LIQUID,SOLID)850
C     ICONDF  = ON-OFF INDICATOR SHOWING WHETHER COND. IS A FUNCTION ANYWHERE 860
C     ITEMPF  = ON-OFF INDICATOR SHOWING WHETHER COND,SPMASS, OR SPHEAT IS FUNC.
C     Q       = HEAT SOURCE AT EACH NODE                                880
C     IQREGN  = 1-2 INDICATOR SHOWING WHETHER Q IS NON-ZERO IN EACH REGION 890
C     IQF     = ON-OFF INDICATOR SHOWING WHETHER Q IS A FUNCTION ANYWHERE 900
C     SPMASS  = DENSITY          FOR EACH MATERIAL BY STATE (LIQUID,SOLID) 910
C     SPHEAT  = SPECIFIC MASS    FOR EACH MATERIAL BY STATE (LIQUID,SOLID) 920
C     IMASSF  = ON-OFF INDICATOR SHOWING WHETHER SPMASS IS A FUNCTION ANYWHERE930
C     IHEATF  = ON-OFF INDICATOR SHOWING WHETHER SPHEAT IS A FUNCTION ANYWHERE940
C     TMELT   = MELTING POINT TEMPERATURE FOR EACH MELTABLE MATERIAL    950
C     FUSION  = LATENT HEAT OF FUSION      FOR EACH MELTABLE MATERIAL   960
C     ENERGY  = LATENT HEAT ENERGY BANK STORED FOR EACH NODE (CHANGE OF STATE)970
C     IMELT   = ON-OFF INDICATOR SHOWING WHETHER ANY MATERIAL CAN MELT ANYWHERE
C     IAMLIQ  = ON-OFF INDICATOR SHOWING WHEN THE FIRST NODE MELTS      990
C     LIQUID  = 1-2 INDICATOR SHOWING STATE OF EACH NODE (1=SOLID, 2=LIQUID) 1000
C     TMPOLD  = TEMPERATURES AT THE PREVIOUS ITERATION                  1010
C     TMPNEW  = TEMPERATURES CALCULATED FOR THE CURRENT ITERATION       1020
C     TEMP    = TMPOLD FOR THE TRANSIENT CASE                           1030
C     AVGTMP  = AVERAGE TEMPERATURE FOR ANY OF SIX DIFFERENT GROUPS     1040
C     NAVGRP  = NUMBER OF GROUPS TO BE AVERAGED                         1050
C     IGRPND  = GROUP NUMBER FOR EACH OF THE SIX GROUPS                 1060
C     NAVGND  = NUMBER OF NODES TO BE AVERAGED IN EACH GROUP (MAXNOD = 60) 1070
C     MAXNOD  = MAXIMUM NUMBER OF NODES THAT CAN BE AVERAGED IN ONE GROUP 1080
C     NODAVG  = NODE NUMBERS OF NODES TO BE AVERAGED IN EACH GROUP      1090
C     FIXTMP  = TEMPERATURE VALUES TO BE FIXED FOR CERTAIN NODES IN MESH 1100
C     NFIXED  = NUMBER OF DIFFERENT TEMPERATURE VALUES TO BE FIXED      1110
C     MAXFIX  = MAXIMUM NUMBER OF DIFFERENT TEMP.S TO BE FIXED          1120
C     NODFIX  = FIXTMP SUBSCRIPT SHOWING WHICH FIXTMP GOES WITH WHICH NODE 1130
C     TDERIV  = TIME DERIVATIVE OF TEMPERATURE (FOR PROPERTIES=FUNCTION(TEMP))
C     ALLMAX  = MAXIMUM TEMPERATURE ALLOWED OVER WHOLE MESH (CHECKED FOR) 1150
C     OUTSID  = MAXIMUM BOUNDARY TEMPERATURE (EXTERNAL OR EDGE) AT EACH STEP 1160
C     TMPMAX  = MAXIMUM TEMPERATURE OVER THE WHOLE MESH                 1170
C     FNAME   = MATERIAL NAME OF THE NODE WHERE TMPMAX WAS MEASURED     1180
C     ITPMAX  = ROW NUMBER OF THE NODE WHERE TMPMAX WAS MEASURED        1190
C     JTPMAX  = COL NUMBER OF THE NODE WHERE TMPMAX WAS MEASURED        1200
C     TMPMIN  = MINIMUN TEMPERATURE OVER THE WHOLE MESH                 1210
C     ITPMIN  = ROW NUMBER OF THE NODE WHERE TMPMIN WAS MEASURED        1220
C     JTPMIN  = COL NUMBER OF THE NODE WHERE TMPMIN WAS MEASURED        1230
C     TCHECK  = MAXIMUM ALLOWABLE TEMPERATURE FOR SPECIFIED NODES (SEE MAXAVG)
C     NCHECK  = NO OF NODES TO BE CHECKED AGAINST TCHECK FOR TEMP. TOO HIGH 1250
C     NODCHK  = INTERNAL NODE NUMBERS OF NODES TO BE CHECKED AGAINST TCHECK 1260
```

```
C       MAXCHK = MAXIMUM NUMBER OF NODES THAT CAN BE CHECKED AGAINST TCHECK (250)
C       IEXCED = ON-OFF INDICATOR SHOWING WHEN A CHECKED NODE EXCEEDS TCHECK  1280
C       TSTEP  = USER'S TIME STEP INCREMENT FOR EACH TIME STEP ITERATION      1290
C       DELTA  = TSTEP/2 = INTERNAL TIME STEP INCREMENT BETWEEN ROW AND COL SEARCH
C       DELTA2 = DELTA/2 = ONE FOURTH USER'S TIME STEP INCREMENT              1310
C       STRTIM = INITIAL START TIME (REAL TIME) WHICH IS NORMALLY ZERO        1320
C       TIME   = ACTUAL ACCUMULATED REAL TIME AT ANY POINT IN INTERNAL TIME UNITS
C       SECTIM = ACTUAL ACCUMULATED REAL TIME AT ANY POINT IN SECONDS         1340
C       INSTEP = INITIAL TIME STEP NUMBER (INTEGER) (NORMALLY 0 UNLESS RESTARTING)
C       NSTEPS = TOTAL NUMBER OF TIME STEPS IN THE PROBLEM                    1360
C       LISTMP = FREQUENCY AT WHICH TEMP.S ARE TO BE PRINTED (GIVEN FOR EACH STEP)
C       ICURR  = CURRENT NODAL ROW NUMBER                                     1380
C       JCURR  = CURRENT NODAL COLUMN NUMBER                                  1390
C       KCURR  = CURRENT NODAL INTERNAL NODE NUMBER                           1400
C       LCURR  = CURRENT NODAL INTERNAL REGION NUMBER                         1410
C       IFTYPE = CURRENT NODAL FACE TYPES                                     1420
C       AREA   = CURRENT NODAL FACE AREAS                                     1430
C       CONDK  = CURRENT NODAL CONDUCTIVITY                                   1440
C       BS     = CURRENT ADJUSTED INTER-NODAL DISTANCES                       1450
C       TMPS   = CURRENT TEMPERATURE OF NEIGHBORING NODES (OR BOUNDARY)       1460
C       QK     = CURRENT NODAL HEAT SOURCE                                    1470
C       TEMOLD = CURRENT NODAL TEMPERATURE (PREVIOUS INCREMENT)               1480
C       SMASSK = CURRENT NODAL DENSITY                                        1490
C       SHEATK = CURRENT NODAL SPECIFIC HEAT                                  1500
C       LIQUDK = CURRENT NODAL STATE INDICATOR (1=SOLID, 2=LIQUID)            1510
C       W      = CURRENT NODAL PARAMETER = CONDK*AREA*DELTA/2                 1520
C       VK     = CURRENT NODAL PARAMETER = SMASSK*SHEATK*VOL                  1530
C       UK     = CURRENT NODAL PARAMETER = QK*VOL*DELTA                       1540
C       BIGW   = CURRENT NODAL PARAMETER = (1 OR 2)*W                         1550
C       INTGER = INTEGER WORK ARRAY USED BY INPUT SUBROUTINES                 1560
C       KNODES = NODE NUMBERS WORK ARRAY USED BY INPUT SUBROUTINES            1570
C       INTDIM = DIMENSION USED FOR INTGER ARRAY                              1580
C       MAXKNO = DIMENSION USED FOR KNODES ARRAY                              1590
C       EXPSET = SET OF 'FOR-EXPRESSION' EXPRESSION NAMES                     1600
C       LOCSET = SET OF 'FOR-EXPRESSION' LOCATION   NAMES                     1610
C       NEXP   = NUMBER OF EXPRESSION NAMES                                   1620
C       NLOC   = NUMBER OF LOCATION   NAMES                                   1630
C       VARNMS = SET OF 'ARITHMETIC-EXPRESSION' VARIABLE NAMES                1640
C       NVAR   = NUMBER OF VARIABLE NAMES                                     1650
C       NUMERC = ON-OFF INDICATOR SHOWING WHEN VARNMS IS NUMERIC OR ALPHAMERIC1660
C       NVAL   = NUMBER OF VALUES TO RHS OF EQUAL SIGN FOR EACH VARNMS        1670
C       VARVAL = RHS VALUES OF EACH 'ARITH.-EXPRESSION'                       1680
C       TITLE  = ALPHAMERIC TITLE CARDS READ IN FOR EACH CASE                 1690
C       A      = LOWER BAND OF ABC THREE BAND MATRIX USED IN SOLVING FOR TEMP.S
C       B      = DIAGONAL   OF ABC THREE BAND MATRIX USED IN SOLVING FOR TEMP.S
C       C      = UPPER BAND OF ABC THREE BAND MATRIX USED IN SOLVING FOR TEMP.S
C       P      = RIGHT HAND SIDE OF THE MATRIX EQUATION, ABC*TEMP=P           1730
C       ABC    = THREE BAND MATRIX (BANDS SHIFTED TO MAKE 3 COL.S) DESCRIBED ABOVE
C*****END VARIABLE DEFINITIONS                                                1750
C                                                                             1760
      IMPLICIT REAL*8 (A-H,O-Z)                                               1770
C*****************************************************************************
C*****BEGIN COMMON AREAS FOR WHOLE PROGRAM ***********************************
C                                                                             1800
C*****BEGIN MAXIMUM DIMENSIONS COMMON AREA                                     1810
      COMMON /GEOM/   MHREG ,MVREG ,MREGS ,MHVREG,                            1820
     1                MHNODE,MVNODE,MNODES,MHVNOD                             1821
      COMMON /MATERL/ MNMATL                                                  1830
      COMMON /ITERAT/ MNITR                                                   1840
C*****END MAXIMUM DIMENSIONS COMMON AREA                                      1850
C                                                                             1860
C*****BEGIN PARAMETER COMMON AREA                                              1870
      COMMON /RERUN/  INITAL                                                  1880
      COMMON /PRINT/  INPRNT,MESHPT,NITRPR,ITRTMP,NLINES                      1890
```

```
      COMMON /SAVE/   ISAVE ,LSAVE
      COMMON /UNITS/  LENGTH,MASS  ,ITIME ,ITEMP  ,IHEAT
      COMMON /ITRATE/ ITR,MAXITR
      COMMON /EXECTM/ MAXSEC,NOWTME,INITME
C*****END PARAMETER COMMON AREA
C
C*****BEGIN MESH GEOMETRY COMMON AREA
      COMMON /CCORD/  IGEOM
      COMMON /IJKMAX/ IMAX,JMAX,KMAX
      COMMON /REGION/ HRDIST(50),VRDIST(50),
     1                NHREG     ,NVREG     ,NTREG,
     2                NHRNDS(50),NVRNDS(50),
     3                JTOLH(1000),ITOLV(1000),
     4                LHTOJ(50,2),LVTOI(50,2)
      INTEGER*2       ITOLV,JTOLH,LVTOI,LHTOJ
      COMMON /MESH/   RDELTA(50),R0,R(1000),RBIG0,RBIG(1000),
     1                ZDELTA(50),Z0,Z(1000),ZBIG0,ZBIG(1000)
      COMMON /FAREAS/ AREA1(1000),RAREA4(1001)
      DIMENSION       AREA3(1000),RAREA2(1000)
      EQUIVALENCE    (AREA1, AREA3),(RAREA4(2),RAREA2(1))
C     AREA4 = RAREA4*ZDELTA
C     AREA2 = RAREA2*ZDELTA
      COMMON /NDVOLS/ VOL(2500),REGVOL(2500),TOTVOL
      COMMON /FACES/  IFT256(4,256),FTYPE(10000)
      LOGICAL*1       FTYPE
      INTEGER*2       IFT256
C*****END MESH GEOMETRY COMMON AREA
C
C*****BEGIN BOUNDARY CONDITIONS COMMON AREA
      COMMON /BNDARY/ TOBAR(4,1000),ARATE(4,1000),IBNDRY(4,1000)
      LOGICAL*1       IBNDRY
      COMMON /CHANGE/ TEMPCH(5000),DISTCH(5000),TINITL(4,1000)
C*****END BOUNDARY CONDITIONS COMMON AREA
C
C*****BEGIN MATERIAL PROPERTIES COMMON AREA
      COMMON /REGMTL/ MTLNAM(300),NMATL,MATLNO(2500)
      REAL*8          MTLNAM
      COMMON /MTCOND/ COND(300,2),ICONDF,ITEMPF
      COMMON /HEATQ/  Q(10000),IQREGN(2500),IQF
      COMMON /SPECIF/ SPMASS(300,2),SPHEAT(300,2),IMASSF,IHEATF
      COMMON /MELTNG/ TMELT(300),FUSION(300),ENERGY(10000),IMELT,
     1                IAMLIQ,LIQUID(10000)
      LOGICAL*1 LIQUID
C*****END MATERIAL PROPERTIES COMMON AREA
C
C*****BEGIN TEMPERATURE COMMON AREA
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)
      DIMENSION       TEMP  (10000)
      EQUIVALENCE    (TMPOLD,TEMP)
C     TEMP = OLD TEMPERATURES AT PREVIOUS TIME STEP FOR TRANSIENT        TRANSENT
      COMMON /AVGTEM/ AVGTMP(6),NAVGRP,IGRPNO(6),NAVGND(6),MAXNOD,
     1                NODAVG(6,60)
C     MAXNOD = 60 = MAXIMUM NUMBER OF NODES AVERAGED IN ONE GROUP
      COMMON /FIXTEM/ FIXTMP(255),NFIXED,MAXFIX,NODFIX(10000)
      LOGICAL*1 NODFIX
C     MAXFIX = 255 = MAXIMUM NO. OF DIFFERENT TEMP.S TO BE FIXED (NODFIX=BYTE)
      COMMON /DERIV/  TDERIV(10000)
      COMMON /MAXTMP/ ALLMAX,OUTSID,TMPMAX,FNAME,ITPMAX,JTPMAX
      COMMON /MINTMP/ TMPMIN,ITPMIN,JTPMIN
      COMMON /CHECKT/ TCHECK(250),NCHECK,NODCHK(250),MAXCHK,IEXCED(250)
      LOGICAL*1 IEXCED
C     MAXCHK = 250 = MAXIMUM NO. OF NODES TO BE CHECKED FOR TEMP.S TOO HIGH
C*****END TEMPERATURE COMMON AREA
C
```

```
C*****BEGIN REAL TIME COMMON AREA
      COMMON /REALTM/ TSTEP(5000),DELTA,DELTA2,STRTIM,TIME,SECTIM,
     1                INSTEP,NSTEPS
      COMMON /LISTNG/ LISTMP(5000)
      INTEGER*2       LISTMP
C*****END REAL TIME COMMON AREA
C
C*****BEGIN NODAL INFORMATION COMMON AREA
      COMMON /NDIJKL/ ICURR,JCURR,KCURR,LCURR
      COMMON /NDFACE/ IFTYPE(4)
      COMMON /NDAREA/ AREA(4)
      COMMON /NDCOND/ CONDK
      COMMON /NDBS/   BS(4)
      COMMON /NDTMPS/ TMPS(4)
      COMMON /NDQ/    QK
      COMMON /TEMPK/  TEMOLD
      COMMON /NDMASS/ SMASSK
      COMMON /NDHEAT/ SHEATK
      COMMON /NDSTAT/ LIQUDK
      COMMON /WUVBIG/ W(4),VK,UK,BIGW(4)
C*****END NODAL INFORMATION COMMON AREA
C
C*****BEGIN WORK ARRAYS COMMON AREA
      COMMON /MAXDIM/ INTDIM,MAXKNO
C     INTDIM =  500 = COLUMN DIMENSION OF THE ARRAY, INTGER
C     MAXKNO = 5000 = DIMENSION OF THE ARRAY, KNODES
      COMMON /WORK1/  INTGER(2,500)
      COMMON /WORK2/  KNODES(5000)
      INTEGER*2       INTGER,KNODES
      COMMON /EXPRES/ EXPSET(19),NEXP
      COMMON /LOCATE/ LOCSET(31),NLOC
      REAL*8          LOCSET
      COMMON /AEXPNM/ VARNMS(30),NVAR,NUMERC(30),NVAL(30)
      COMMON /HVDIST/ VARVAL(50)
      COMMON /HEADNG/ TITLE(10,12)
      COMMON /BANDS/  A(1000),B(1000),C(1000),P(1000)
      DIMENSION     ABC(1000,3)
C     EQUIVALENCE  (ABC(1,1),A(1)),(ABC(1,2),B(1)),(ABC(1,3),C(1))
      EQUIVALENCE  (ABC(1,1),A(1))
C*****END WORK ARRAYS COMMON AREA
C
C*****END COMMON AREAS FOR WHOLE PROGRAM ***************************************
C*******************************************************************************
C
C*****BEGIN INPUT AND INITIALIZATION
      CALL INITLZ
 1    CALL INPUT
      IF(MESHPT .EQ. 1) CALL MESHPR
      INITME=ICLOCK(0)/100
C     CALL SPACNG  IS DONE INSIDE OF SUBROUTINE INPUT
C     CALL AREAS   IS DONE INSIDE OF SUBROUTINE INPUT
C     CALL VOLUME  IS DONE INSIDE OF SUBROUTINE INPUT
      CALL NDTYPE
      CALL TMPOUT(-1)
      CALL STATE
C*****END INPUT AND INITIALIZATION
C
C*****BEGIN OUTER TIME STEP LOOP ***********************************************
      TIME=STRTIM
      IREFPT=INSTEP-1
      DO 37 ITR=INSTEP,NSTEPS
C*****BEGIN TIME CALCULATIONS
      DELTA=TSTEP(ITR)/2.0D+00
      DELTA2=DELTA/2.0D+00
```

```
      TIME=TIME+TSTEP(ITR)
      IF(ITIME .EQ. 1) SECTIM=TIME
      IF(ITIME .EQ. 2) SECTIM=TIME*60.0D+00
      IF(ITIME .EQ. 3) SECTIM=TIME*3600.0D+00
C*****END TIME CALCULATIONS
C*****BEGIN BOUNDARY CHANGE AS FUNCTION OF TIME
      IF(TEMPCH(ITR) .GT. (-9999998.0D+00)) CALL TOBARF(TEMPCH(ITR))
      IF(DISTCH(ITR) .GT. (-9999998.0D+00)) CALL ARATEF(DISTCH(ITR))
C*****END BOUNDARY CHANGE AS FUNCTION OF TIME
C
      ISWEEP=0
 2    ISWEEP=ISWEEP+1
C*****BEGIN COLUMN-BY-COLUMN MESH SWEEP
      DO 11 J=1,JMAX
      DO 8 I=1,IMAX
C*****BEGIN NODE DESCRIPTION CALCULATIONS
      K=(I-1)*JMAX+J
      LV=ITOLV(I)
      LH=JTOLH(J)
      L=(LV-1)*NHREG+LH
      MATL=MATLNO(L)
      ICURR=I
      JCURR=J
      KCURR=K
      LCURR=L
C     I,J,K    = NODE   POSITION
C     LV,LH,L  = REGION POSITION
C     MATL     = NODE   MATERIAL
C*****END NODE DESCRIPTION CALCULATIONS
C*****BEGIN NODAL PROPERTY CALCULATIONS
      LIQUDK=LIQUID(K)
      CONDK =GTCOND(MATL,LIQUDK)
      SMASSK=GTMASS(MATL,LIQUDK)
      SHEATK=GTHEAT(MATL,LIQUDK)
      QK     =GETQK (K)
      CALL   GETYPE(K)
      CALL   GTAREA(I,J)
      CALL   GETBS (I,J)
      CALL   GETMPS(I,J,K)
      TEMOLD=TMPOLD(K)
C*****END NODAL PROPERTY CALCULATIONS
C*****BEGIN ABCP MATRIX CALCULATIONS
      CONDEL=CONDK*DELTA2
      DO 3 IS=1,4
 3    W(IS)=AREA(IS)*CONDEL
      BIGW(1)=W(1)
      BIGW(2)=W(2)*2.0D+00
      BIGW(3)=W(3)
      BIGW(4)=W(4)*2.0D+00
      VK=SMASSK*SHEATK*VOL(L)
      UK=QK*VOL(L)*DELTA
      W1B1=0.0D+00
      W3B3=0.0D+00
      IF(BS(1) .NE. 0.0D+00) W1B1=W(1)/BS(1)
      IF(BS(3) .NE. 0.0D+00) W3B3=W(3)/BS(3)
      A(I)=-W1B1
      B(I)=W1B1+W3B3+VK
      C(I)=-W3B3
      P(I)=UK+VK*TEMOLD
      IF(I .NE. 1) GO TO 4
      A(I)=0.0D+00
      P(I)=P(I)+W1B1*TMPS(1)
      GO TO 5
 4    IF(I .NE. IMAX) GO TO 5
```

```
      C(I)=0.0D+00
      P(I)=P(I)+W3B3*TMPS(3)
C*****BEGIN ADDITIONAL CALC.S FOR FULLY IMPLICIT ALTERNATING DIRECTION METHOD
 5    ITEMPF=1
      BIGW(2)=W(2)
      BIGW(4)=W(4)
      W2B2=0.0D+00
      W4B4=0.0D+00
      IF(BS(2) .NE. 0.0D+00) W2B2=W(2)/BS(2)
      IF(BS(4) .NE. 0.0D+00) W4B4=W(4)/BS(4)
      B(I)=B(I)+W2B2+W4B4
      IF(J .EQ. JMAX) P(I)=P(I)+W2B2*TMPS(2)
      IF(J .EQ. 1) P(I)=P(I)+W4B4*TMPS(4)
      IF((ISWEEP .EQ. 1) .AND. (J .NE. JMAX))
     1   P(I)=P(I)+(TDERIV(K+1)*DELTA+TMPOLD(K+1))*W2B2
      IF((ISWEEP .EQ. 2) .AND. (J .NE. JMAX))
     1   P(I)=P(I)+(TMPNEW(K+1))*W2B2
      IF(J .NE. 1) P(I)=P(I)+(TMPNEW(K-1))*W4B4
C*****END ADDITIONAL CALC.S FOR FULLY IMPLICIT ALTERNATING DIRECTION METHOD
 6    DO 7 IS=1,4
      IF(BS(IS) .EQ. 0.0D+00) GO TO 7
      P(I)=P(I)+(BIGW(IS)/BS(IS))*(TMPS(IS)-TEMOLD)
 7    CONTINUE
C*****END ABCP MATRIX CALCULATIONS
 8    CONTINUE
C*****BEGIN SOLUTION OF MATRIX FOR TEMP.S (EXCEPT FIXED TEMP. NODES)
      CALL BANDSO(IMAX,3)
      DO 10 I=1,IMAX
      K=(I-1)*JMAX+J
      IFIX=NODFIX(K)
      IF(IFIX .EQ. 0) GO TO 9
      TMPNEW(K)=FIXTMP(IFIX)
      GO TO 10
 9    TMPNEW(K)=P(I)
 10   CONTINUE
C*****END SOLUTION OF MATRIX FOR TEMP.S (EXCEPT FIXED TEMP. NODES)
 11   CONTINUE
C*****END COLUMN-BY-COLUMN MESH SWEEP
C*****BEGIN NODE MELTING CHECK
      IF((IMELT .NE. 0) .AND. (ISWEEP .EQ. 2)) CALL MELT
C*****END NODE MELTING CHECK
C*****BEGIN TEMP. DERIVATIVE
      IF(ITEMPF .EQ. 0) GO TO 13
      DO 12 K=1,KMAX
 12   TDERIV(K)=(TMPNEW(K)-TMPOLD(K))/DELTA
 13   CONTINUE
C*****END TEMP. DERIVATIVE
C*****BEGIN BOUNDARY CHANGE AS A FUNCTION OF TIME
      IF(ISWEEP .EQ. 2) GO TO 14
      IF(TEMPCH(ITR) .GT. (-9999998.0D+00)) CALL TOBARF(TEMPCH(ITR))
      IF(DISTCH(ITR) .GT. (-9999998.0D+00)) CALL ARATEF(DISTCH(ITR))
 14   CONTINUE
C*****END BOUNDARY CHANGE AS A FUNCTION OF TIME
      IF(ISWEEP .EQ. 1) GO TO 2
C
C*****BEGIN TEMP. RESETTING
      DO 15 K=1,KMAX
 15   TMPOLD(K)=TMPNEW(K)
C*****END TEMP. RESETTING
C
C
C
C*****BEGIN BOUNDARY CHANGE AS A FUNCTION OF TIME
      IF(TEMPCH(ITR) .GT. (-9999998.0D+00)) CALL TOBARF(TEMPCH(ITR))
```

```
      IF(DISTCH(ITR) .GT. (-9999998.0D+00)) CALL ARATEF(DISTCH(ITR))
C*****END BOUNDARY CHANGE AS A FUNCTION OF TIME
C
      ISWEEP=0
 16   ISWEEP=ISWEEP+1
C*****BEGIN ROW-BY-ROW MESH SWEEP
      DO 25 I=1,IMAX
      DO 22 J=1,JMAX
C*****BEGIN NODE DESCRIPTION CALCULATIONS
      K=(I-1)*JMAX+J
      LV=ITOLV(I)
      LH=JTOLH(J)
      L=(LV-1)*NHREG+LH
      MATL=MATLNO(L)
      ICURR=I
      JCURR=J
      KCURR=K
      LCURR=L
C     I,J,K    = NODE   POSITION
C     LV,LH,L  = REGION POSITION
C     MATL     = NODE   MATERIAL
C*****END NODE DESCRIPTION CALCULATIONS
C****BEGIN NODAL PROPERTY CALCULATIONS
      LIQUDK=LIQUID(K)
      CONDK =GTCOND(MATL,LIQUDK)
      SMASSK=GTMASS(MATL,LIQUDK)
      SHEATK=GTHEAT(MATL,LIQUDK)
      QK     =GETQK(K)
      CALL   GETYPE(K)
      CALL   GTAREA(I,J)
      CALL   GETBS(I,J)
      CALL   GETMPS(I,J,K)
      TEMOLD=TMPOLD(K)
C*****END NODAL PROPERTY CALCULATIONS
C*****BEGIN ABCP MATRIX CALCULATIONS
      CONDEL=CONDK*DELTA2
      DO 17 IS=1,4
 17   W(IS)=AREA(IS)*CONDEL
      BIGW(1)=W(1)*2.0D+00
      BIGW(2)=W(2)
      BIGW(3)=W(3)*2.0D+00
      BIGW(4)=W(4)
      VK=SMASSK*SHEATK*VOL(L)
      UK=QK*VOL(L)*DELTA
      W4B4=0.0D+00
      W2B2=0.0D+00
      IF(BS(4) .NE. 0.0D+00) W4B4=W(4)/BS(4)
      IF(BS(2) .NE. 0.0D+00) W2B2=W(2)/BS(2)
      A(J)=-W4B4
      B(J)=W4B4+W2B2+VK
      C(J)=-W2B2
      P(J)=UK+VK*TEMOLD
      IF(J .NE. 1) GO TO 18
      A(J)=0.0D+00
      P(J)=P(J)+W4B4*TMPS(4)
      GO TO 19
 18   IF(J .NE. JMAX) GO TO 19
      C(J)=0.0D+00
      P(J)=P(J)+W2B2*TMPS(2)
C*****BEGIN ADDITIONAL CALC.S FOR FULLY IMPLICIT ALTERNATING DIRECTION METHOD
 19   ITEMPF=1
      BIGW(1)=W(1)
      BIGW(3)=W(3)
      W1B1=0.0D+00
      W3B3=0.0D+00
      IF(BS(1) .NE. 0.0D+00) W1B1=W(1)/BS(1)
```

```
      IF(BS(3) .NE. 0.0D+00) W3B3=W(3)/BS(3)
      B(J)=B(J)+W1B1+W3B3
      IF(I .EQ. IMAX) P(J)=P(J)+W3B3*TMPS(3)
      IF(I .EQ. 1) P(J)=P(J)+W1B1*TMPS(1)
      IF((ISWEEP .EQ. 1) .AND. (I .NE. IMAX))
     1   P(J)=P(J)+(TDERIV(K+JMAX)*DELTA+TMPOLD(K+JMAX))*W3B3
      IF((ISWEEP .EQ. 2) .AND. (I .NE. IMAX))
     1   P(J)=P(J)+(TMPNEW(K+JMAX))*W3B3
      IF(I .NE. 1) P(J)=P(J)+(TMPNEW(K-JMAX))*W1B1
C*****END ADDITIONAL CALC.S FOR FULLY IMPLICIT ALTERNATING DIRECTION METHOD
 20   DO 21 IS=1,4
      IF(BS(IS) .EQ. 0.0D+00) GO TO 21
      P(J)=P(J)+(BIGW(IS)/BS(IS))*(TMPS(IS)-TEMOLD)
 21   CONTINUE
C*****END ABCP MATRIX CALCULATIONS
 22   CONTINUE
C*****BEGIN SOLUTION OF MATRIX FOR TEMP.S (EXCEPT FIXED TEMP.NODES)
      CALL BANDSO(JMAX,3)
      DO 24 J=1,JMAX
      K=(I-1)*JMAX+J
      IFIX=NODFIX(K)
      IF(IFIX .EQ. 0) GO TO 23
      TMPNEW(K)=FIXTMP(IFIX)
      GO TO 24
 23   TMPNEW(K)=P(J)
 24   CONTINUE
C*****END SOLUTION OF MATRIX FOR TEMP.S (EXCEPT FIXED TEMP. NODES)
 25   CONTINUE
C*****END ROW-BY-ROW MESH SWEEP
C*****BEGIN NODE MELTING CHECK
      IF(IMELT .EQ. 0) GO TO 26
      IF(ISWEEP .EQ. 1) GO TO 26
      CALL MELT
 26   CONTINUE
C*****END NODE MELTING CHECK
C*****BEGIN TEMP. DERIVATIVE
      IF(ITEMPF .EQ. 0) GO TO 28
      DO 27 K=1,KMAX
 27   TDERIV(K)=(TMPNEW(K)-TMPOLD(K))/DELTA
 28   CONTINUE
C*****END TEMP. DERIVATIVE
C*****BEGIN BOUNDARY CHANGE AS A FUNCTION OF TIME
      IF(ISWEEP .EQ. 2) GO TO 29
      IF(TEMPCH(ITR) .GT. (-9999998.0D+00)) CALL TOBARF(TEMPCH(ITR))
      IF(DISTCH(ITR) .GT. (-9999998.0D+00)) CALL ARATEF(DISTCH(ITR))
 29   CONTINUE
C*****END BOUNDARY CHANGE AS A FUNCTION OF TIME
      IF(ISWEEP .EQ. 1) GO TO 16
C
C*****BEGIN MAX., MIN., AND AVG. TEMP. CALCULATIONS
      CALL MAXAVG
C*****END MAX., MIN., AND AVG. TEMP. CALCULATIONS
C*****BEGIN ITERATION OUTPUT
      NOWTME=ICLOCK(0)/100-INITME
      IF(ITR .EQ. 1) GO TO 30
      IF(NITRPR .EQ. 0) GO TO 31
      IDIV=ITR/NITRPR
      INEW=IDIV*NITRPR
      IF(INEW .NE. ITR) GO TO 31
 30   CALL ITROUT
 31   IF(ITR .EQ. INSTEP) GO TO 32
      IF(LISTMP(ITR) .NE. LISTMP(ITR-1)) IREFPT=ITR-1
 32   ITFREQ=LISTMP(ITR)
      IF(ITFREQ .EQ. 0) GO TO 33
```

```
      LOCITR=ITR-IREFPT
      IDIV=LOCITR/ITFREQ
      INEW=IDIV*ITFREQ
      IF(INEW .NE. LCCITR) GO TO 33
      IF(ITRTMP .EQ. 0) CALL MAPOUT(0)
      IF(ITRTMP .EQ. 1) CALL TMPOUT(0)
 33   CONTINUE
C*****END ITERATICN CUTPUT
C*****BEGIN MACHINE EXECUTION TIME CHECK
      IF(MAXSEC .EQ. 0) GO TO 35
      IF(NOWTME .LT. MAXSEC) GO TO 35
      INEW=0
      IF(NITRPR .EQ. 0) GO TO 34
      IDIV=ITR/NITRPR
      INEW=IDIV*NITRPR
 34   IF(INEW .NE. ITR) CALL ITROUT
      PRINT 1001,NOWTME,MAXSEC,ITR
 1001 FORMAT(//1X,'MACHINE TIME =',I7,' SEC.S HAS EXCEEDED MAXIMUM ALLOW
     1ABLE TIME =',I7,' SEC.S AFTER TIME STEP =',I6//)
      NLINES=NLINES+5
      GO TO 38
 35   CONTINUE
C*****END MACHINE EXECUTION TIME CHECK
C*****BEGIN RESETTING TEMP.S
      DO 36 K=1,KMAX
 36   TMPOLD(K)=TMPNEW(K)
C*****END RESETTING TEMP.S
 37   CONTINUE
      ITR=NSTEPS
C*****END OUTER TIME STEP LOOP ***********************************************
C
C*****BEGIN FINAL CUTPUT
 38   CALL MAPOUT(1)
      CALL TMPOUT(1)
C*****END FINAL OUTPUT
C*****BEGIN FINAL SAVE OUTPUT
      IF(ISAVE .EQ. 0) GO TO 39
      CALL TPSAVE
C*****END FINAL SAVE OUTPUT
 39   PRINT 1002
 1002 FORMAT(1H1)
      GO TO 1
      END
```

APPENDIX E

LISTING OF SKELETON SUBPROGRAM FOR CONDF (STEADY-STATE ONLY)

```
      FUNCTION CONDF(MATL)
      IMPLICIT REAL*8 (A-H,O-Z)
C*****************************************************************
C     THIS IS A DUMMY CONDUCTIVITY FUNCTION WHICH MAY SERVE AS A MODEL
C     FOR A USER-WRITTEN FUNCTION. IT RETURNS A SINGLE REAL*8 VALUE
C     REPRESENTING THE CONDUCTIVITY AT THE FACE, ISIDE, OF NODE, K.
C     WE ASSUME THAT THE CONDUCTIVITY IS A FUNCTION OF TEMPERATURE HERE
C     SO THAT WE NEED TO KNOW THE TEMPERATURE ON THE FACE. ONE EASY WAY
C     TO APPROXIMATE THIS TEMPERATURE IS TO AVERAGE THE TEMPERATURES ON
C     EACH SIDE OF THE FACE..
C
C          TEMP(FACE)  =  (TEMP(K) + TEMP(NEIGHBOR))/2
C
C     THIS RESULTS IN QUICK CALCULATIONS. THE WAY WE DO IT HERE IS MORE
C     TIME-CONSUMING BUT IS MORE ACCURATE. WE APPROXIMATE THE TEMPERATURE
C     BY..
C
C          TEMP(FACE)  =   TEMP(K) + DELTAX * DT/DN
C          DT/DN       =  (TEMP(NEIGHBOR) - TEMP(K)) / DIST
C
C     OF COURSE IF THE NODES ARE EQUALLY SPACED, THE TWO TECHNIQUES ARE ALMOST
C     THE SAME. THUS THE CONDUCTIVITY MAY BE CALCULATED ONCE THE TEMP(FACE)
C     IS KNOWN FROM AN EQUATION OR TABLE LOOKUP. IF A USER WANTED TO
C     REALLY SPEED UP CALCULATIONS HE COULD LET..
C
C          TEMP(FACE)   =   TEMP(K)
C
C     OF COURSE THIS IS VERY INACCURATE FOR STEEP TEMPERATURE GRADIENTS.
C*****************************************************************
C*****BEGIN COMMON AREAS
      COMMON /COORD/  IGEOM
      COMMON /IJKMAX/ IMAX,JMAX,KMAX
      COMMON /MESH/   RDELTA(50),R0,R(1000),RBIG0,RBIG(1000),
     1                ZDELTA(50),Z0,Z(1000),ZBIG0,ZBIG(1000)
      COMMON /FACES/  IFT256(4,256),FTYPE(10000)
      LOGICAL*1       FTYPE
      INTEGER*2       IFT256
      COMMON /BNDARY/ TOBAR(4,1000),ARATE(4,1000),IBNDRY(4,1000)
      LOGICAL*1       IBNDRY
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)
      DIMENSION       TEMP  (10000)
      EQUIVALENCE    (TMPNEW,TEMP)
C     TEMP = BEST (NEWEST) TEMPERATURES FOR STEADY STATE
      COMMON /NDIJKL/ ICURR,JCURR,KCURR,LCURR
      COMMON /NDCOND/ CONDK,ISIDE
      DATA ALPHA/0.1D+00/
C*****END COMMON AREAS
C*****BEGIN INITIALIZATION
      I=ICURR
      J=JCURR
      K=KCURR
      IF(ISIDE .NE. 0) GO TO 1
      CALL MISQU(2,'CONDF CALLED WITH ISIDE=0 - COND=F(TEMP) BUT NO FACE
     1 FOR K,MATL=$',K,MATL)
      CONDF=900000.0D+00
      RETURN
```

```
C*****END INITIALIZATION                                                COND 550
C*****BEGIN FINDING TYPE OF FACE, ISIDE                                  COND 560
 1      I256=FTYPE(K)                                                   COND 570
        ITYPE=IFT256(ISIDE,I256)                                        COND 580
C*****END FINDING TYPE OF FACE, ISIDE                                   COND 590
C*****BEGIN GETTING NEIGHBORING TEMPERATURE                             COND 600
        GO TO (2,2,7,7), ITYPE                                          COND 610
 2      GO TO (3,4,5,6), ISIDE                                          COND 620
 3      TSIDE=TEMP(K-JMAX)                                              COND 630
        GO TO 12                                                        COND 640
 4      TSIDE=TEMP(K+1)                                                 COND 650
        GO TO 12                                                        COND 660
 5      TSIDE=TEMP(K+JMAX)                                              COND 670
        GO TO 12                                                        COND 680
 6      TSIDE=TEMP(K-1)                                                 COND 690
        GO TO 12                                                        COND 700
 7      G  TO (8,9,10,11), ISIDE                                        COND 710
 8      TSIDE=TOBAR(1,J)                                                COND 720
        GO TO 12                                                        COND 730
 9      TSIDE=TOBAR(2,I)                                                COND 740
        GO TO 12                                                        COND 750
 10     TSIDE=TOBAR(3,J)                                                COND 760
        GO TO 12                                                        COND 770
 11     TSIDE=TOBAR(4,I)                                                COND 780
        GO TO 12                                                        COND 790
 12     CONTINUE                                                        COND 800
C*****END GETTING NEIGHBORING TEMPERATURE                               COND 810
C*****BEGIN GETTING WITHIN-NODAL DISTANCE, DELTAX                       COND 820
        CONS=1.0D+00                                                    COND 830
        IF(IGEOM .EQ. 3) CONS=RBIG(J)                                   COND 840
        GO TO (13,14,15,16), ISIDE                                      COND 850
 13     DELTAX=CONS*(ZBIG(I)-Z(I-1))                                    COND 860
        GO TO 17                                                        COND 870
 14     DELTAX=R(J)-RBIG(J)                                             COND 880
        GO TO 17                                                        COND 890
 15     DELTAX=CONS*(Z(I)-ZBIG(I))                                      COND 900
        GO TO 17                                                        COND 910
 16     DELTAX=RBIG(J)-R(J-1)                                           COND 920
        GO TO 17                                                        COND 930
 17     CONTINUE                                                        COND 940
C*****END GETTING WITHIN-NODAL DISTANCE, DELTAX                         COND 950
C*****BEGIN GETTING BETWEEN-NODAL DISTANCE, DIST                        COND 960
        ARECIP=0.0D+00                                                  COND 970
        IF(ITYPE .NE. 3) GO TO 18                                       COND 980
        DIST=0.0D+00                                                    COND 990
        ISUB=I                                                          CON 1000
        IF((ISIDE .EQ. 1) .OR. (ISIDE .EQ. 3)) ISUB=J                   CON 1010
        A=ARATE(ISIDE,ISUB)                                             CON 1020
        IF(A .EQ. 0.0D+00) GO TO 23                                     CON 1030
        ARECIP=1.0D+00/A                                                CON 1040
 18     ZKONE=1.0D+00                                                   CON 1050
        IF((ITYPE .EQ. 3) .OR. (ITYPE .EQ. 4)) ZKONE=0.0D+00            CON 1060
        GO TO (19,20,21,22), ISIDE                                      CON 1070
 19     DIST=DELTAX + CONS*(Z(I-1)-ZBIG(I-1))*ZKONE + ARECIP            CON 1080
        GO TO 23                                                        CON 1090
 20     DIST=DELTAX +       (RBIG(J+1)-R(J)   )*ZKONE + ARECIP          CON 1100
        GO TO 23                                                        CON 1110
```

```
 21   DIST=DELTAX + CONS*(ZBIG(I+1)-Z(I)   )*ZKONE + ARECIP                CON 1120
      GO TO 23                                                             CON 1130
 22   DIST=DELTAX +       (R(J-1)-RBIG(J-1))*ZKONE + ARECIP                CON 1140
      GO TO 23                                                             CON 1150
 23   CONTINUE                                                             CON 1160
C*****END GETTING BETWEEN-NODAL DISTANCE, DIST                             CON 1170
C*****BEGIN CALCULATION OF TEMP(FACE)                                      CON 1180
      IF(DIST .NE. 0.0D+00) GO TO 24                                       CON 1190
      TFACE=TMPNEW(K)                                                      CON 1200
      GO TO 25                                                             CON 1210
 24   DTDN = (TSIDE - TMPNEW(K)) / DIST                                    CON 1220
      TFACE = TMPNEW(K) + DELTAX * DTDN                                    CON 1230
 25   CONTINUE                                                             CON 1240
C*****END CALCULATION OF TEMP(FACE)                                        CON 1250
C                                                                          CON 1260
C*****BEGIN USER-WRITTEN FUNCTION OF THE TEMPERATURE, TFACE                CON 1270
C                                                                          CON 1280
      CONDF = 1.0D+00 + ALPHA * TFACE                                      CON 1290
C                                                                          CON 1300
C     THE ABOVE FUNCTION SHOULD BE REPLACED BY WHATEVER FUNCTION THE USER HAS.
C*****END USER-WRITTEN FUNCTION OF THE TEMPERATURE, TFACE                  CON 1320
      CALL MISQU(2,'CONDF CALLED - THIS IS A DUMMY FUNCTION - MATL,K=$',CON 1330
     1           MATL,K)                                                   CON 1331
C     REMOVE THE ABOVE ERROR CALL WHEN A FUNCTION IS SUPPLIED.             CON 1340
      RETURN                                                               CON 1350
      END                                                                  CON 1360
```

LISTING OF SKELETON SUBPROGRAM FOR CONDF (TRANSIENT ONLY)

```
      FUNCTION CONDF(MATL,LIQUDK)                                       CONDF 10
      IMPLICIT REAL*8 (A-H,O-Z)                                         CONDF 20
C*********************************************************************
C     THIS IS A DUMMY CONDUCTIVITY FUNCTION WHICH MAY BE OVER-RIDDEN BY CONDF 40
C     A USER WRITTEN FUNCTION. (IT KEEPS THE LINKAGE EDITOR HAPPY.)     CONDF 50
C     HOWEVER, THIS FUNCTION DOES SHOW HOW TO USE THE TEMPERATURE       CONDF 60
C     DERIVATIVE (WITH RESPECT TO TIME) TO ESTIMATE THE TEMPERATURE AT  CONDF 70
C     THE CURRENT TIME STEP IN CASE THE CONDUCTIVITY IS A FUNCTION OF   CONDF 80
C     TEMPERATURE. IT IS ASSUMED THAT THE CURRENT NODE NUMBER, K, IS    CONDF 90
C     STORED IN THE COMMON VARIABLE, KCURR. IT IS ALSO ASSUMED THAT     COND 100
C     THE TEMPERATURE DERIVATIVE HAS BEEN CALCULATED FOR THIS NODE AT   COND 110
C     THE PREVICUS TIME STEP, TIME-DELTA. TIME IS THE CURRENT TIME AND  COND 120
C     DELTA IS THE CURRENT TIME INTERVAL. TMPOLD(KCURR) IS THE TEMPERATURE  130
C     AT THE PREVICUS TIME STEP. NOTE THAT DELTA = TSTEP(ITR)/2 SO THAT THE  140
C     USER TIME STEPS, TSTEP, ARE ACTUALLY CUT IN HALF AND USED AS THE  COND 150
C     INTERNAL TIME STEPS, DELTA. THE EQUATION IS..                     COND 160
C                                                                       COND 170
C          TMPEST = TMPOLD(K) + TDERIV(K)*DELTA                         COND 180
C                                                                       COND 190
C*********************************************************************
C*****BEGIN COMMON AREAS                                                COND 210
      COMMON /NDIJKL/ ICURR,JCURR,KCURR,LCURR                           COND 220
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)                       COND 230
      DIMENSION       TEMP  (10000)                                     COND 240
      EQUIVALENCE    (TMPOLD,TEMP)                                      COND 250
C     TEMP = OLD TEMPERATURES AT PREVIOUS TIME STEP FOR TRANSIENT       TRANSENT
      COMMON /DERIV/  TDERIV(10000)                                     COND 270
      COMMON /REALTM/ TSTEP(5000),DELTA,DELTA2,STRTIM,TIME,SECTIM,      COND 280
     1                INSTEP,NSTEPS                                     COND 281
C*****END COMMON AREAS                                                  COND 290
      TMPEST = TMPOLD(KCURR) + TDERIV(KCURR)*DELTA                      COND 300
 1    CALL MISQU(3,'CONDF ERR1 - CONDUCTIVITY FUNCTION CALLED - MATL,LIQCOND 310
     1UDK,KCURR=$',MATL,LIQUDK,KCURR)                                   COND 311
      CONDF=9000000.0D+00                                               COND 320
      RETURN                                                            COND 330
      END                                                               COND 340
```

LISTING OF SKELETON SUBPROGRAM FOR SHEATF (TRANSIENT ONLY)

```
      FUNCTION  SHEATF(MATL,LIQUDK)                                     SHEAT 10
      IMPLICIT  REAL*8 (A-H,O-Z)                                        SHEAT 20
C*****************************************************************************
C     THIS FUNCTION IS IDENTICAL TO FUNCTION CONDF EXCEPT THAT THE      SHEAT 40
C     SPECIFIC HEAT IS BEING CALCULATED INSTEAD OF THE CONDUCTIVITY.    SHEAT 50
C     THIS FUNCTICN SHOWS HOW TO ESTIMATE THE TEMPERATURE JUST LIKE     SHEAT 60
C     CONDF DOES. SEE FUNCTION CONDF FOR A DETAILED DESCRIPTION.        SHEAT 70
C*****************************************************************************
C*****BEGIN COMMON AREAS                                                SHEAT 90
      COMMON /NDIJKL/ ICURR,JCURR,KCURR,LCURR                           SHEA 100
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)                       SHEA 110
      DIMENSION       TEMP  (10000)                                     SHEA 120
      EQUIVALENCE    (TMPOLD,TEMP)                                      SHEA 130
C     TEMP = OLD TEMPERATURES AT PREVIOUS TIME STEP FOR TRANSIENT       TRANSENT
      COMMON /DERIV/  TDERIV(10000)                                     SHEA 150
      COMMON /REALTM/ TSTEP(5000),DELTA,DELTA2,STFTIM,TIME,SECTIM,      SHEA 160
     1                INSTEP,NSTEPS                                    SHEA 161
C*****END COMMCN AREAS                                                  SHEA 170
      TMPEST = TMPOLD(KCURR) + TDERIV(KCURR)*DELT/                      SHEA 180
 1    CALL MISQU(3,'SHEATF ERR1 - SPECIFIC HEAT FLNCTION CALLED - MATL,LSHEA 190
     1LIQUDK,KCURR=$',MATL,LIQUDK,KCURR)                                SHEA 191
      SHEATF=9000000.0D+00                                              SHEA 200
      RETURN                                                            SHEA 210
      END                                                               SHEA 220
```

LISTING OF SKELETON SUBPROGRAM FOR SMASSF (TRANSIENT ONLY)

```
      FUNCTION  SMASSF(MATL,LIQUDK)                                     SMASS 10
      IMPLICIT  REAL*8 (A-H O-Z)                                        SMASS 20
C*********************************************************************
C     THIS FUNCTION IS IDENTICAL TO FUNCTION CONDF EXCEPT THAT THE      SMASS 40
C     DENSITY IS BEING CALCULATED INSTEAD OF THE CONDUCTIVITY.          SMASS 50
C     THIS FUNCTION SHOWS HOW TO ESTIMATE THE TEMPERATURE JUST LIKE     SMASS 60
C     CONDF DOES. SEE FUNCTION CONDF FOR A DETAILED DESCRIPTION.        SMASS 70
C*********************************************************************
C*****BEGIN COMMON AREAS                                                SMASS 90
      COMMON /NDIJKL/ ICURR,JCURR,KCURR,LCURR                           SMAS 100
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)                       SMAS 110
      DIMENSION       TEMP  (10000)                                     SMAS 120
      EQUIVALENCE    (TMPOLD,TEMP)                                      SMAS 130
C     TEMP = OLD TEMPERATURES AT PREVIOUS TIME STEP FOR TRANSIENT       TRANSENT
      COMMON /DERIV/  TDERIV(10000)                                     SMAS 150
      COMMON /REALTM/ TSTEP(5000),DELTA,DELTA2,STRTIM,TIME,SECTIM,      SMAS 160
     1                INSTEP,NSTEPS                                     SMAS 161
C*****END COMMON AREAS                                                  SMAS 170
      TMPEST = TMPOLD(KCURR) + TDERIV(KCURR)*DELTA                      SMAS 180
 1    CALL MISQU(3,'SMASSF ERR1 - SMASSF DENSITY  FUNCTION CALLED - MATLSMAS 190
     1,LIQUDK,KCURR=$',MATL,LIQUDK,KCURR)                               SMAS 191
      SMASSF=9000000.0D+00                                              SMAS 200
      RETURN                                                            SMAS 210
      END                                                               SMAS 220
```

LISTING OF SKELETON SUBPROGRAM FOR QF (TRANSIENT AND STEADY-STATE)

```
      FUNCTION QF(K)                                                    QF     10
      IMPLICIT REAL*8 (A-H,O-Z)                                         QF     20
C*********************************************************************
C     THIS IS A DUMMY HEAT SOURCE FUNCTION WHICH MAY BE REPLACED BY A   QF     40
C     USER-WRITTEN FUNCTION. THIS FUNCTION KEEPS THE LINKAGE EDITOR HAPPY     50
C*********************************************************************
      CALL MISQU(1,'QF CALLED - THIS IS A DUMMY FUNCTION - K=$',K)      QF     70
      QF=9000000.0D+00                                                  QF     80
      RETURN                                                            QF     90
      END                                                               QF    100
```

LISTING OF SKELETON SUBPROGRAM FOR TOBARF (TRANSIENT ONLY)

```
      SUBROUTINE TOBARF(TMPCH)                                          TOBAR 10
      IMPLICIT REAL*8 (A-H,O-Z)                                         TOBAR 20
C*********************************************************************
C     THIS SUBROUTINE IS USED TO ALTER THE BOUNDARY CONDITIONS (SPECIFICALLY  40
C     THE BOUNDARY TEMPERATURE) AS A FUNCTION OF TIME. THE USER SPECIFIES     50
C     A TEMP CHANGE VALUE WHICH IS SENT TO THIS SUBROUTINE AT THE SPECIFIED   60
C     TIME STEP OR STEPS. THIS TEMP CHANGE VALUE MAY BE USED IN ANY WAY THE   70
C     USER DESIRES (BY WRITING TOBARF OR ALTERING THIS ONE) TO CHANGE   TOBAR 80
C     WHATEVER BOUNDARY NODE TEMPERATURES HE WISHES. THIS TEMP CHANGE   TOBAR 90
C     VALUE MAY HAVE UNITS (SUCH AS 'F. OR 'F/S.) ATTACHED TO IT AND    TOBA 100
C     ANY NECESSARY CONVERSION TO INTERNAL UNITS WILL BE DONE BY SUBROUTINE  110
C     INPUT. ALTHOUGH SUBROUTINE TOBARF MAY CHANGE THE ARRAY, TOBAR, IT TOBA 120
C     SHOULD NOT ALTER THE TYPE OF BOUNDARY CONDITION (FIXED TEMP. OR   TOBA 130
C     GRADIENT) SPECIFIED IN THE ARRAY, IBNDRY (3=GRADIENT, 4=FIXED TEMP).   140
C     IF IT DOES, THEN SUBROUTINE NDTYPE SHOULD BE CALLED AGAIN HERE.   TOBA 150
C     THERE IS NO WAY IN THE INPUT TO SPECIFY WHICH BOUNDARY NODES ARE  TOBA 160
C     TO BE AFFECTED AT EACH TIME STEP. THIS MUST BE DONE WITHIN TOBARF TOBA 170
C                                                                       TOBA 180
C     THIS PARTICULAR SUBROUTINE SIMPLY TAKES THE TEMP CHANGE VALUE     TOBA 190
C     (ASSUMED TO BE TEMPERATURE) AND PUTS IT INTO EVERY BOUNDARY NODE, TOBA 200
C     TOBAR, DEPENDING UPON THE GEOMETRY USED (R,THETA HAS ONLY SIDE 2  TOBA 210
C     AS A BOUNDARY). IF THE BOUNDARY TEMPERATURE CHANGED LINEARLY,     TOBA 220
C     FOR EXAMPLE, WITH TIME THE USER MIGHT LET THE TEMP CHANGE VALUE   TOBA 230
C     BE A RATE = 'F/S. THEN HE COULD SET THE INITIAL BOUNDARY TEMPERATURE   240
C     WITH THE EXT.TEMP OR EDGE.TEMP INPUT CARD AND CALCULATE HIS BOUNDARY   250
C     TEMPERATURE IN TOBARF WITH AN EQUATION LIKE..                     TOBA 260
C                                                                       TOBA 270
C          TOBAR(ISIDE,NODE) = TINITL(ISIDE,NODE) + TMPCH*TIME          TOBA 280
C                                                                       TOBA 290
C     NOTE THAT TINITL ALWAYS CONTAINS THE INITIAL TEMPERATURES GIVEN ONTOBA 300
C     THE EXT.TEMP OR EDGE.TEMP INPUT CARDS. IN USING THE EQUATION ABOVETOBA 310
C     THE USER ASSUMES THAT THE SLOPE, TMPCH, DOES NOT CHANGE WITH TIME.TOBA 320
C     AS AN ADDED FEATURE IN THIS SUBROUTINE, THE ABOVE PROCEDURE DESCRIBED  330
C     FOR A LINEARLY CHANGING BOUNDARY TEMPERATURE HAS BEEN INCLUDED. ALL    340
C     OF THE VALID BOUNDARY NODES WILL BE CHANGED IN THIS MANNER. TO USETOBA 350
C     THE LINEAR FUNCTION RATHER THAN THE REPLACEMENT PROCEDURE DESCRIBED    360
C     FIRST, SIMPLY CHANGE THIS DATA STATEMENT..                        TOBA 370
C                                                                       TOBA 380
      DATA IUSE/1/                                                      TOBA 390
C                                                                       TOBA 400
C     TO ONE WHICH LOOKS LIKE..                                         TOBA 410
C                                                                       TOBA 420
C     DATA IUSE/2/                                                      LIN FUNC
C                                                                       TOBA 440
C     BY SIMPLY MOVING THE 'C' IN COLUMN 1 FROM THE   ' FUNC CARD TO THETOBA 450
C     REPLCMNT CARD. IN SOME CASES THE USER MAY WISH . VARY THE SLOPE   TOBA 460
C     IN THE LINEAR FUNCTION PROCEDURE WITH TIME TO REPRESENT A NON-LINEAR   470
C     BOUNDARY TEMPERATURE. THIS HAS NOT BEEN CODED HERE BUT CAN BE BY  TOBA 480
C     USING AN EQUATION OF THE FORM..                                   TOBA 490
C                                                                       TOBA 500
C          TOBAR(ISIDE,NODE) = TOBAR(ISIDE,NODE) + TMPCH*(TIME - OLDTIM)TOBA 510
C                                                                       TOBA 520
C     WHERE OLDTIM IS THE TIME SAVED FROM THE LAST CALL TO TOBARF. THE  TOBA 530
C     CODING OF THIS PROCEDURE IS LEFT AS AN EXERCISE FOR THE STUDENT.  TOBA 540
C     HE MIGHT LET ILSE TAKE ON A VALUE OF 3 FOR THIS CASE. (HINT)      TOBA 550
C*********************************************************************
```

```
C*****BEGIN COMMON AREA                                                 TOBA 570
      COMMON /ITRATE/ ITR,MAXITR                                        TOBA 580
      COMMON /COORD/  IGEOM                                             TOBA 590
      COMMON /IJKMAX/ IMAX,JMAX,KMAX                                    TOBA 600
      COMMON /REGION/ HRDIST(50),VRDIST(50),                            TOBA 610
     1                NHREG     ,NVREG     ,NTREG,                      TOBA 611
     2                NHRNDS(50),NVRNDS(50),                            TOBA 612
     3                JTOLH(1000),ITOLV(1000),                          TOBA 613
     4                LHTOJ(50,2),LVTOI(50,2)                           TOBA 614
      INTEGER*2       ITOLV,JTOLH,LVTOI,LHTOJ                           TOBA 620
      COMMON /BNDARY/ TOBAR(4,1000),ARATE(4,1000),IBNDRY(4,1000)        TOBA 630
      LOGICAL*1       IBNDRY                                            TOBA 640
      COMMON /CHANGE/ TEMPCH(5000),DISTCH(5000),TINITL(4,1000)          TOBA 650
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)                       TOBA 660
      DIMENSION       TEMP  (10000)                                     TOBA 670
      EQUIVALENCE    (TMPOLD,TEMP)                                      TOBA 680
C     TEMP = OLD TEMPERATURES AT PREVIOUS TIME STEP FOR TRANSIENT       TRANSENT
      COMMON /REALTM/ TSTEP(5000),DELTA,DELTA2,STRTIM,TIME,SECTIM,      TOBA 700
     1                INSTEP,NSTEPS                                     TOBA 701
C*****END COMMON AREA                                                   TOBA 710
C                                                                       TOBA 720
C*****BEGIN REPLCMNT OR LIN FUNC PROCEDURE                              TOBA 730
      DO 5 ISIDE=1,4                                                    TOBA 740
      IF((IGEOM .EQ. 1) .AND. (ISIDE .EQ. 4)) GO TO 5                   TOBA 750
      IF((IGEOM .EQ. 3) .AND. (ISIDE .NE. 2)) GO TO 5                   TOBA 760
      IF((ISIDE .EQ. 1) .OR.  (ISIDE .EQ. 3)) IJMAX=JMAX                TOBA 770
      IF((ISIDE .EQ. 2) .OR.  (ISIDE .EQ. 4)) IJMAX=IMAX                TOBA 780
      GO TO (1,3), IUSE                                                 TOBA 790
C*****BEGIN REPLCMNT                                                    TOBA 800
 1    DO 2 IJ=1,IJMAX                                                   TOBA 810
 2    TOBAR(ISIDE,IJ)=TMPCH                                             TOBA 820
      GO TO 5                                                           TOBA 830
C*****END REPLCMNT                                                      TOBA 840
C*****BEGIN LIN FUNC                                                    TOBA 850
 3    DO 4 IJ=1,IJMAX                                                   TOBA 860
 4    TOBAR(ISIDE,IJ)=TINITL(ISIDE,IJ)+TMPCH*TIME                       TOBA 870
      GO TO 5                                                           TOBA 880
C*****END LIN FUNC                                                      TOBA 890
 5    CONTINUE                                                          TOBA 900
C*****END REPLCMNT OR LIN FUNC PROCEDURE                                TOBA 910
      RETURN                                                            TOBA 920
      END                                                               TOBA 930
```

LISTING OF SKELETON SUBPROGRAM FOR ARATEF (TRANSIENT ONLY)

```
      SUBROUTINE ARATEF(DISCH)                                          ARATE 10
      IMPLICIT REAL*8 (A-H,O-Z)                                         ARATE 20
C*********************************************************************
C     THIS SUBROUTINE IS USED TO ALTER THE BOUNDARY CONDITIONS (SPECIFICALLY, 40
C     THE GRADIENT DISTANCE) AS A FUNCTION OF TIME. THE ARRAY, ARATE,   ARATE 50
C     CONTAINS THE RECIPROCAL GRADIENT DISTANCE (1/FEET, 1/INCHES, ETC.)ARATE 60
C     FOR ALL FOUR SIDES OF THE BOUNDARY. THIS GRADIENT DISTANCE, WHICH ARATE 70
C     IS ALSO KNOWN AS THE BOUNDARY LAYER THICKNESS, IS JUST            ARATE 80
C     THE CONDUCTIVITY OF THE SURFACE MATERIAL (K=B/FH'F) DIVIDED BY    ARATE 90
C     THE HEAT TRANSFER RATE (H=B/FFH'F) ACROSS THE BOUNDARY. THUS      ARAT 100
C     ARATE = H/K. THIS SUBROUTINE CAN BE CALLED AT ANY TIME STEP TO    ARAT 110
C     CHANGE ANY VALUES IN THE ARATE ARRAY. THIS IS DONE BY INCLUDING   ARAT 120
C     A 'DIST.CHANGE' CARD IN THE INPUT DATA WITH A NUMERICAL VALUE TO  ARAT 130
C     BE SENT TO ARATEF AT THE SPECIFIED TIME STEPS. FOR EXAMPLE,       ARAT 140
C                                                                       ARAT 150
C          DIST.CHANGE  =  0.6 F.  FOR  TIM.STEPS  30-60 .              ARAT 160
C                                                                       ARAT 170
C     THIS VALUE (0.6 F.) MIGHT BE USED AS A NEW GRADIENT DISTANCE FOR  ARAT 180
C     THE TIME STEPS 30-60. AS YOU MIGHT NOTICE, THERE IS NO WAY IN THE ARAT 190
C     'DIST.CHANGE' CARD TO SPECIFY THE BOUNDARY NODES AFFECTED. THIS ISARAT 200
C     DONE BY THE USER WITHIN THIS SUBROUTINE. THE USER MAY ALSO USE THEARAT 210
C     NUMERICAL VALUE IN ANY WAY HE WISHES WITHIN THIS SUBROUTINE.      ARAT 220
C     FOR THE CONVENCIENCE OF THE USER, WE HAVE WRITTEN THIS PARTICULAR ARAT 230
C     SUBROUTINE TO DO ONE OF TWO THINGS DEPENDING UPON THE VALUE OF    ARAT 240
C     THE VARIABLE, IUSE. IUSE MAY BE SET EQUAL TO 1 OR 2 BY USING      ARAT 250
C     THE APPROPRIATE DATA STATEMENT GIVEN AT THE END OF THIS COMMENTARY.    260
C                                                                       ARAT 270
C     IF IUSE IS SET EQUAL TO 1, THEN A STRAIGHT REPLACEMENT PROCEDURE IS    280
C     USED. THIS PROCEDURE SIMPLY REPLACES THE OLD ARATE VALUE IN EVERY ARAT 290
C     BOUNDARY NODE WITH THE NEW ONE SENT TO THIS SUBROUTINE VIA THE    ARAT 300
C     ARGUMENT LIST (DISCH). ACTUALLY, THE RECIPROCAL OF DISCH (THE     ARAT 310
C     GRADIENT DISTANCE) IS PUT INTO ARATE. ALL BOUNDARY NODES FOR THE  ARAT 320
C     PARTICULAR GEOMETRY BEING USED ARE CHANGED. IN OUR PREVIOUS EXAMPLE,   330
C     THE NEW ARATE VALUE WOULD BE 1.666 1/FEET. (ARATE IS DIMENSIONED  ARAT 340
C     BY SIDE AND NODE, ARATE(ISIDE,NODE).)                             ARAT 350
C                                                                       ARAT 360
C     IF IUSE IS SET EQUAL TO 2, THEN A RADIATION-CONVECTION PROCEDURE  ARAT 370
C     IS USED. THIS PROCEDURE ALLOWS RADIATION PLUS FORCED CONVECTION   ARAT 380
C     PLUS NATURAL CONVECTION TO OCCUR AT ALL OUTSIDE BOUNDARIES WHICH  ARAT 390
C     ARE NOT INSULATED OR SET AT A FIXED TEMPERATURE INITIALLY. THUS   ARAT 400
C     THOSE NODES WHICH ARE 'INSULATED' OR GIVEN AN 'EDGE.TEMP' INITIALLY    410
C     ARE NOT INVOLVED IN THE RADIATION-CONVECTION PROCEDURE. THE       ARAT 420
C     EQUATION WHICH EXPRESSES RADIATION PLUS FORCED AND NATURAL CONVECTION  430
C     IS GIVEN BELOW IN TERMS OF AN EFFECTIVE HEAT TRANSFER COEFFICIENT, H.. 440
C                                                                       ARAT 450
C          H       = HCONV  +                                           FORCED
C                    HNATRL * ABS( (TB-TEXT)/TCONS )**HE  +             NATURAL
C                    STEFAN * EMISS * (TB**4 - TEXT**4)/(TB - TEXT)     RADIATE
C                                                                       ARAT 490
C     OR IN TERMS OF THE ARATE VARIABLE..                               ARAT 500
C                                                                       ARAT 510
C          ARATE   = ACONV +                                            FORCED
C                    HNATRL*1/CONDK * ABS( (TB-TEXT)/TCONS )**HE +      NATURAL
C                    STEFAN*EMISS*1/CONDK * (TB**4 - TEXT**4)/(TB-TEXT) RADIATE
C                                                                       ARAT 550
C     WHERE THE TERMS ARE..                                             ARAT 560
C                                                                       ARAT 570
C          HCONV   = HEAT TRANSFER RATE DUE TO FORCED CONVECTION        B/FFH'F
C          ACONV   = ARATE DUE TO FORCED CONVECTION = HCONV/CONDK       1/FEET
C          HNATRL  = NATURAL CONVECTION COEFFICIENT                     B/FFH'F
C          HE      = EXPONENT FOR THE TERM SIMULATING NATURAL CONVECTION UNITLESS
C          STEFAN  = STEFAN-BOLTZMANN CONSTANT          = 5.6697D-08 W/MM'K'K'K'K
C          EMISS   = EMISSIVITY OF THE SURFACE                          UNITLESS
```

```
C          CONDK   = CONDUCTIVITY OF THE SURFACE MATERIAL                B/FH'F
C          TB      = SURFACE TEMP OF BOUNDARY AT NEXT TIME STEP          'F    650
C          TEXT    = EXTERNAL TEMPERATURE OF THE OUTSIDE MEDIUM          'F    660
C          TCONS   = A CONSTANT TEMP VALUE OF ONE DEGREE FAHRENHEIT      1.0 'F
C                                                                        ARAT 680
C     THE ACONV VALUE ABOVE IS JUST 1/DISCH = 1/FEET = 1/0.6 IN OUR      ARAT 690
C     PREVIOUS EXAMPLE. THUS THE USER CAN SPECIFY THE FORCED CONVECTION ARAT 700
C     CONDITION ON HIS 'DIST.CHANGE' INPUT CARD. (TO IGNORE FORCED       ARAT 710
C     CONVECTION SIMPLY SET THE NUMERICAL VALUE TO 0.0 ON THE 'DIST.CHANGE'  720
C     CARD. THE 'DIST.CHANGE' CARD SHOULD STILL SPECIFY WHEN THIS SUBROUTINE 730
C     IS TO BE CALLED). THE OTHER THREE PARAMETERS, HNATRL, HE, AND EMISS,   740
C     MUST BE SPECIFIED WITHIN THIS SUBROUTINE IN THE DATA STATEMENTS    ARAT 750
C     GIVEN AT THE END OF THIS COMMENTARY. IF NATURAL CONVECTION IS TO   ARAT 760
C     BE IGNORED, THEN HNATRL SHOULD BE SET TO ZERO. LIKEWISE, THE       ARAT 770
C     EMISSIVITY SHOULD BE SET TO ZERO IF RADIATION IS NOT ALLOWED.      ARAT 780
C     NOTE THAT WHEN FAHRENHEIT DEGREES ARE BEING USED, THAT THE NATURALARAT 790
C     CONVECTION TERM LOOKS JUST LIKE THE STANDARD TEXTBOOK EQUATIONS.. ARAT 800
C                                                                        ARAT 810
C          H(TURBULENT) = HNATRL*ABS(TB-TEXT)**1/3                       ARAT 820
C          H(LAMINAR)   = HNATRL*ABS(TB-TEXT)**1/4                       ARAT 830
C                                                                        ARAT 840
C     THE HNATRL VALUE AND ITS CORRESPONDING UNITS FIELD ARE WRITTEN IN THE  850
C     DATA STATEMENT BELOW AS ONE LONG HOLLERITH STRING. THE UNITS USED ARAT 860
C     MAY BE DIFFERENT THAN THE INTERNAL UNITS GIVEN IN THE HEADER SECTION.  870
C     THE REST OF THE ABOVE PARAMETERS ARE KNOWN EXCEPT FOR TB. IT MUST ARAT 880
C     BE FOUND BY SOLVING A FOURTH DEGREE EQUATION USING THE NEWTON-RAPHSON  890
C     METHOD. THIS METHOD CALCULATES TB AS SHOWN BELOW..                 ARAT 900
C                                                                        ARAT 910
C          TB(I+1) = TB(I) -  F(TB(I)) / F'(TB(I))                       ARAT 920
C                                                                        ARAT 930
C     WHERE F(TB) IS THE FOURTH DEGREE EQUATION IN TB. IN THIS EQUATION ARAT 940
C     WE MUST APPROXIMATE THE OUTER NODAL TEMPERATURE AT THE NEXT TIME STEP..950
C                                                                        ARAT 960
C          TNEW(K) = TOLD(K) + TDERIV(K)*DELTA                           ARAT 970
C                                                                        ARAT 980
C     IN ALL OF THESE CALCULATIONS THE TEMP.S MUST BE CONVERTED TO KELVIN,   990
C     AND STEFAN TO THE INTERNAL UNITS. THE FINAL RESULT WILL BE AN ARATE   1000
C     VALUE FOR EACH BOUNDARY NODE.                                      ARA 1010
C                                                                        ARA 1020
      LOGICAL*1 HNATRL(30)                                               ARA 1030
C     DATA IUSE  /1/                                                     REPLACE
      DATA IUSE  /2/                                                     ARA 1050
      DATA HNATRL/3CH  0.19D+00 B/FFH'F.          /                      ARA 1060
      DATA HE    /     0.333333333D+00            /                      ARA 1070
      DATA EMISS /     0.9                        /                      ARA 1080
C                                                                        ARA 1090
C***********************************************************************
C*****BEGIN COMMON AREA                                                  ARA 1110
      COMMON /UNITS/  LENGTH, MASS  ,ITIME ,ITEMP ,IHEAT                 ARA 1120
      COMMON /ITRATE/ ITR,MAXITR                                         ARA 1130
      COMMON /COORD/  IGEOM                                              ARA 1140
      COMMON /IJKMAX/ IMAX,JMAX,KMAX                                     ARA 1150
      COMMON /REGION/ HRDIST(50),VRDIST(50),                             ARA 1160
     1                NHREG     ,NVREG     ,NTREG,                       ARA 1161
     2                NHRNDS(50),NVRNDS(50),                             ARA 1162
     3                JTOLH(1000),ITOLV(1000),                           ARA 1163
     4                LHTOJ(50,2),LVTOI(50,2)                            ARA 1164
      INTEGER*2       ITOLV,JTOLH,LVTOI,LHTOJ                            ARA 1170
      COMMON /MESH/   RDELTA(50),R0,R(1000),RBIG0,RBIG(1000),            ARA 1180
     1                ZDELTA(50),Z0,Z(1000),ZBIG0,ZBIG(1000)             ARA 1181
      COMMON /BNDARY/ TOBAR(4,1000),ARATE(4,1000),IBNDRY(4,1000)         ARA 1190
      LOGICAL*1       IBNDRY                                             ARA 1200
      COMMON /CHANGE/ TEMPCH(5000),DISTCH(5000),TINITL(4,1000)           ARA 1210
      COMMON /REGMTL/ MTLNAM(300),NMATL,MATLNO(2500)                     ARA 1220
```

```
      REAL*8            MTLNAM                                          ARA 1230
      COMMON /MTCCND/ COND(300,2),ICCNDF,ITEMPF                         ARA 1240
      COMMON /MELTNG/ TMELT(300),FUSION(300),ENERGY(10000),IMELT,       ARA 1250
     1                IAMLIQ,LIQUID(10000)                              ARA 1251
      LOGICAL*1 LIQLID                                                  ARA 1260
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)                       ARA 1270
      DIMENSION       TEMP  (10000)                                     ARA 1280
      EQUIVALENCE    (TMPOLD,TEMP)                                      ARA 1290
C     TEMP = OLD TEMPERATURES AT PREVIOUS TIME STEP FOR TRANSIENT       TRANSENT
      COMMON /DERIV/  TDERIV(10000)                                     ARA 1310
      COMMON /REALTM/ TSTEP(5000),DELTA,DELTA2,STRTIM,TIME,SECTIM,      ARA 1320
     1                INSTEP,NSTEPS                                     ARA 1321
      COMMON /NDIJKL/ ICURR,JCURR,KCURR,LCURR                           ARA 1330
      COMMON /NDFACE/ IFTYPE(4)                                         ARA 1340
      COMMON /NDCCND/ CONDK                                             ARA 1350
      COMMON /NDSTAT/ LIQUDK                                            ARA 1360
C*****END COMMON AREA                                                   ARA 1370
C*****BEGIN CONVERSION CONSTANTS                                        ARA 1380
      DIMENSION CLNGTH(4,4),CTIME(3,3),CHEAT(5,5),CTEMP(4,4),TPCONS(4,4)ARA 1390
      DATA CLNGTH/      1.0D+00,     12.0D+00,.3936992D+00,39.36992D+00, ARA 1400
     1            .0833333D+00,      1.0D+00,.0328083D+00, 3.28083D+00, ARA 1401
     2              2.54001D+00,30.48006D+00,      1.0D+00,    100.0D+00, ARA 1402
     3            .0254001D+00,.0304800D+00,      .01D+00,      1.0D+00/ ARA 1403
      DATA CTIME /        1.0D+00,      60.0D+00,    3600.0D+00,         ARA 1410
     1            .01666666D+00,       1.0D+00,      60.0D+00,          ARA 1411
     2            .00027777770D+00,.01666666D+00,       1.0D+00/         ARA 1412
      DATA CHEAT /        1.0D+00,.396587D-02,.9478672D-03,.9478672D-10,ARA 1420
     1.9478672D-03, 252.15105D+00,     1.0D+00,.2390057D+00,.2390057D-07,ARA 1421
     2.2390057D+00,     1055.0D+00, 4.1840D+00,      1.0D+00,      1.0D-07,ARA 1422
     3     1.0D+00,     1055.0D+07, 4.1840D+07,      1.0D+07,      1.0D+00,ARA 1423
     4     1.0D+07,     1055.0D+00, 4.1840D+00,      1.0D+00,      1.0D-07,ARA 1424
     5     1.0D+00/                                                     ARA 1425
      DATA CTEMP /        1.0D+00,       1.0D+00,1.8D+00,1.8D+00,        ARA 1430
     1                    1.0D+00,       1.0D+00,1.8D+00,1.8D+00,        ARA 1431
     2            .55555555D+00,.55555555D+00,1.0D+00,1.0D+00,           ARA 1432
     3            .55555555D+00,.55555555D+00,1.0D+00,1.0D+00/           ARA 1433
      DATA TPCONS/        0.0D+00,-459.67D+00,  32.0D+00,-459.67D+00,   ARA 1440
     1                459.67D+00,     0.0D+00,491.67D+00,     0.0D+00,  ARA 1441
     2            -17.777777D+00,-273.15D+00,    0.0D+00,-273.15D+00,   ARA 1442
     3            255.372222D+00,     0.0D+00,273.15D+00,     0.0D+00/  ARA 1443
C*****END CONVERSION CONSTANTS                                          ARA 1450
C*****BEGIN SPECIFICATIONS                                              ARA 1460
      COMMON /RCDATA/ LUNIT,LENGTH,  ISKIP,ISEP,       IPRINT,IERROR,   ARA 1470
     1                INITNO,NCARDS, NCHAR,CHAR(256), IPOINT,CARD(1024) ARA 1471
      LOGICAL*1 CHAR,CARD                                               ARA 1480
      LOGICAL*1 CHARZ(256),CARDZ(1024)                                  ARA 1490
      LOGICAL*1 LENGUN(4)/'IFCM'/,BLANK1/1H /                           ARA 1500
C*****END SPECIFICATIONS                                                ARA 1510
      DATA IYET/0/                                                      ARA 1520
C                                                                       ARA 1530
      IF(IUSE .EQ. 2) GO TO 3                                           ARA 1540
      IF(DISCH .EQ. 0.0D+00) GO TO 18                                   ARA 1550
      RECIP=1.00+00/DISCH                                               ARA 1560
C*****BEGIN REPLCMNT PROCEDURE                                          ARA 1570
      DO 2 ISIDE=1,4                                                    ARA 1580
      IF((IGEOM .EQ. 1) .AND. (ISIDE .EQ. 4)) GO TO 2                   ARA 1590
      IF((IGEOM .EQ. 3) .AND. (ISIDE .NE. 2)) GO TO 2                   ARA 1600
      IF((ISIDE .EQ. 1) .OR.  (ISIDE .EQ. 3)) IJMAX=JMAX                ARA 1610
      IF((ISIDE .EQ.2) .OR.  (ISIDE .EQ. 4)) IJMAX=IMAX                 ARA 1620
      DO 1 IJ=1,IJMAX                                                   ARA 1630
 1    ARATE(ISIDE,IJ)=RECIP                                             ARA 1640
 2    CONTINUE                                                          ARA 1650
C*****END REPLCMNT PROCEDURE                                            ARA 1660
      RETURN                                                            ARA 1670
```

```
C*****BEGIN RADIATION PLUS CONVECTION AROUND ALL BOUNDARIES                ARA 1680
 3    ITEMPF=1                                                            ARA 1690
C*****BEGIN HNATRL CONVERSION TO INTERNAL UNITS                            ARA 1700
      LUNITZ=LUNIT                                                        ARA 1710
      LENGTZ=LENGT1                                                       ARA 1720
      ISKIPZ=ISKIP                                                        ARA 1730
      ISEPZ =ISEP                                                         ARA 1740
      IPRINZ=IPRINT                                                       ARA 1750
      IERROZ=IERROR                                                       ARA 1760
      INITNZ=INITNO                                                       ARA 1770
      NCARDZ=NCARDS                                                       ARA 1780
      NCHARZ=NCHAR                                                        ARA 1790
      DO 4 I=1,256                                                        ARA 1800
      CHARZ(I)=CHAR(I)                                                    ARA 1810
 4    CHAR(I)=BLANK1                                                      ARA 1820
      IPOINZ=IPOINT                                                       ARA 1830
      DO 5 I=1,1024                                                       ARA 1840
      CARDZ(I)=CARD(I)                                                    ARA 1850
 5    CARD(I)=BLANK1                                                      ARA 1860
      LUNIT=99                                                            ARA 1870
      LENGT1=30                                                           ARA 1880
      NCHAR=0                                                             ARA 1890
      IPOINT=1                                                            ARA 1900
      DO 6 I=1,30                                                         ARA 1910
 6    CARD(I)=HNATRL(I)                                                   ARA 1920
      HNATUL=GETVAL(CLDVAL,OLDIMN,VALDIM,NDCHAR)                          ARA 1930
      LUNIT =LUNITZ                                                       ARA 1940
      LENGT1=LENGTZ                                                       ARA 1950
      ISKIP =ISKIPZ                                                       ARA 1960
      ISEP  =ISEPZ                                                        ARA 1970
      IPRINT=IPRINZ                                                       ARA 1980
      IERROR=IERROZ                                                       ARA 1990
      INITNO=INITNZ                                                       ARA 2000
      NCARDS=NCARDZ                                                       ARA 2010
      NCHAR =NCHARZ                                                       ARA 2020
      DO 7 I=1,256                                                        ARA 2030
 7    CHAR(I)=CHARZ(I)                                                    ARA 2040
      IPOINT=IPOINZ                                                       ARA 2050
      DO 8 I=1,1024                                                       ARA 2060
 8    CARD(I)=CARDZ(I)                                                    ARA 2070
C*****END HNATRL CONVERSION TO INTERNAL UNITS                             ARA 2080
      IF(IYET .EQ. 0) PRINT 1001,ITR,LENGUN(LENGTH),DISCH,VALDIM,HNATUL,ARA 2090
     1                                HE,EMISS                             ARA 2091
 1001 FORMAT(/////' BOUNDARY CONDITIONS ARE CHANGED FOR THE FIRST TIME AARA 2100
     1T TIME STEP',I6/1X,'NEW BOUNDARY LAYER THICKNESS FOR FORCED CONVECARA 2101
     2TION (',A1,'.)      =',F14.6/1X,'NEW NATURAL CONVECTION COEFFICIENTARA 2102
     3 (',A8,')                =',F14.6/1X,'NEW NATURAL CONVECTION EXPONEARA 2103
     4NT (UNITLESS)                =',F14.6/1X,'NEW EMISSIVITY VALUE FARA 2104
     5OR RADIATION (UNITLESS)                =',F14.6///)                 ARA 2105
      IYET=1                                                              ARA 2110
      DO 17 ISIDE=1,4                                                     ARA 2120
      IF((IGEOM .EQ. 1) .AND. (ISIDE .EQ. 4)) GO TO 17                    ARA 2130
      IF((IGEOM .EQ. 3) .AND. (ISIDE .NE. 2)) GO TO 17                    ARA 2140
      GO TO (9,10,11,12), ISIDE                                           ARA 2150
 9    I1=1                                                                ARA 2160
      I2=1                                                                ARA 2170
      J1=1                                                                ARA 2180
      J2=JMAX                                                             ARA 2190
      GO TO 13                                                            ARA 2200
 10   I1=1                                                                ARA 2210
      I2=IMAX                                                             ARA 2220
      J1=JMAX                                                             ARA 2230
      J2=JMAX                                                             ARA 2240
      GO TO 13                                                            ARA 2250
```

```
 11    I1=IMAX                                                          ARA 2260
       I2=IMAX                                                          ARA 2270
       J1=1                                                             ARA 2280
       J2=JMAX                                                          ARA 2290
       GO TO 13                                                         ARA 2300
 12    I1=1                                                             ARA 2310
       I2=IMAX                                                          ARA 2320
       J1=1                                                             ARA 2330
       J2=1                                                             ARA 2340
C*****BEGIN MOVING ALONG BOUNDARY SIDE                                  ARA 2350
 13    IJ=0                                                             ARA 2360
       DO 16 I=I1,I2                                                    ARA 2370
       DO 16 J=J1,J2                                                    ARA 2380
C*****BEGIN NODE DESCRIPTION CALCULATIONS                               ARA 2390
       IJ=IJ+1                                                          ARA 2400
       K=(I-1)*JMAX+J                                                   ARA 2410
       LV=ITOLV(I)                                                      ARA 2420
       LH=JTOLH(J)                                                      ARA 2430
       L=(LV-1)*NHREG+LH                                                ARA 2440
       MATL=MATLNO(L)                                                   ARA 2450
       ICURR=I                                                          ARA 2460
       JCURR=J                                                          ARA 2470
       KCURR=K                                                          ARA 2480
       LCURR=L                                                          ARA 2490
C      I,J,K    = NODE   POSITION                                       ARA 2500
C      LV,LH,L  = REGION POSITION                                       ARA 2510
C      MATL     = NODE   MATERIAL                                       ARA 2520
C*****END NODE DESCRIPTION CALCULATIONS                                 ARA 2530
C*****BEGIN NODAL PROPERTY CALCULATIONS                                 ARA 2540
       CALL   GETYPE(K)                                                 ARA 2550
       IF((IFTYPE(ISIDE) .EQ. 3) .AND. (ARATE(ISIDE,IJ) .EQ. 0.0D+00))  ARA 2560
      1   GO TO 16                                                      ARA 2561
C      IF NODE IS INSULATED, IGNORE IT                                  ARA 2570
       IF(IFTYPE(ISIDE) .EQ. 4) GO TO 16                                ARA 2580
C      IF EDGE TEMP GIVEN, IGNORE IT                                    ARA 2590
       IF(IFTYPE(ISIDE) .NE. 3) GO TO 19                                ARA 2600
       LIQUDK=LIQUID(K)                                                 ARA 2610
       CONDK =GTCOND(MATL,LIQUDK)                                       ARA 2620
       TNODE=TMPOLD(K)                                                  ARA 2630
       TEXT=TOBAR(ISIDE,IJ)                                             ARA 2640
       STEFAN=5.6697D-08                                                ARA 2650
       ACONV=0.0D+00                                                    ARA 2660
       IF(DISCH .NE. 0.0D+00) ACONV=1.0D+00/DISCH                       ARA 2670
       TDERV=TDERIV(K)                                                  ARA 2680
C*****END NODAL PROPERTY CALCULATIONS                                   ARA 2690
C*****BEGIN UNITS CONVERSION                                            ARA 2700
       IFAREN=1                                                         ARA 2710
       KELVIN=4                                                         ARA 2720
C                                                                       ARA 2730
       IBTU=1                                                           ARA 2740
       JOULE=3                                                          ARA 2750
C                                                                       ARA 2760
       ISEC=1                                                           ARA 2770
       IHOUR=3                                                          ARA 2780
C                                                                       ARA 2790
       IFEET=2                                                          ARA 2800
       METER=4                                                          ARA 2810
C                                                                       ARA 2820
       TKELVN=CTEMP(ITEMP,KELVIN)                                       ARA 2830
       ABSOLU=TPCONS(ITEMP,KELVIN)                                      ARA 2840
       TNODE=TNODE*TKELVN+ABSOLU                                        ARA 2850
       TEXT=TEXT*TKELVN+ABSOLU                                          ARA 2860
       TDERV=TDERV*TKELVN                                               ARA 2870
       STEFAN=STEFAN*CHEAT(JOULE,IHEAT)                                 ARA 2880
```

```
      STEFAN=STEFAN*CTIME(ITIME,ISEC)                                   ARA 2890
      STEFAN=STEFAN*(CLNGTH(LENGTH,METER)**2)                           ARA 2900
      CONDU=CONDK*CTEMP(KELVIN,ITEMP)                                   ARA 2910
      ANATUL=HNATUL/CONDK                                               ARA 2920
C*****END UNITS CONVERSION                                              ARA 2930
C*****BEGIN NODE TEMP ESTIMATE AT NEXT TIME STEP                         ARA 2940
      TMPEST = TNODE + TDERV*DELTA                                      ARA 2950
C*****END NODE TEMP ESTIMATE AT NEXT TIME STEP                          ARA 2960
C*****BEGIN NEWTON-RAPHSON METHOD TO CALCULATE TB                       ARA 2970
      TGUESS=TMPEST                                                     ARA 2980
      TOL=0.0001D+00                                                    ARA 2990
      IF(ISIDE .EQ. 1) DELX=ZBIG(1)                                     ARA 3000
      IF(ISIDE .EQ. 2) DELX=R(JMAX)-RBIG(JMAX)                          ARA 3010
      IF(ISIDE .EQ. 3) DELX=Z(IMAX)-ZBIG(IMAX)                          ARA 3020
      IF(ISIDE .EQ. 4) DELX=RBIG(1)                                     ARA 3030
      C=EMISS*STEFAN*DELX/CONDU                                         ARA 3040
      D=ACONV*DELX                                                      ARA 3050
      E=ANATUL*DELX                                                     ARA 3060
      D1=1.0D+00 + D                                                    ARA 3070
      RHSCON=(-TMPEST-C*(TEXT**4)-D*TEXT)                               ARA 3080
 14   TDIFF=TGUESS-TEXT                                                 ARA 3090
      FAHREN=DABS(TDIFF)*CTEMP(KELVIN,IFAREN)                           ARA 3100
      HEPOWR=FAHREN**HE                                                 ARA 3110
      TBETTR = TGUESS - ( C*(TGUESS**4) + D1*TGUESS +                   ARA 3120
     1                    E*HEPOWR*TDIFF + RHSCON ) /                   ARA 3121
     2                  ( 4.0D+00*C*(TGUESS**3) + D1 +                  ARA 3122
     3                    HE*E*(FAHREN**(HE-1.0D+00))*TDIFF +           ARA 3123
     4                    E*HEPOWR )                                    ARA 3124
      IF(DABS(TBETTR-TGUESS) .LE. TOL) GO TO 15                         ARA 3130
      TGUESS=TBETTR                                                     ARA 3140
      GO TO 14                                                          ARA 3150
 15   TB=TBETTR                                                         ARA 3160
C*****END NEWTON-RAPHSON METHOD TO CALCULATE TB                         ARA 3170
C*****BEGIN CALCULATION OF ARATE                                        ARA 3180
      FAHREN=DABS(TB-TEXT)*CTEMP(KELVIN,IFAREN)                         ARA 3190
      ARATE(ISIDE,IJ) = ACONV + ANATUL*(FAHREN**HE) +                   ARA 3200
     1                  STEFAN*EMISS*(TB+TEXT)*(TB**2 + TEXT**2)/CONDU  ARA 3201
C*****END CALCULATION OF ARATE                                          ARA 3210
 16   CONTINUE                                                          ARA 3220
C*****END MOVING ALONG BOUNDARY SIDE                                    ARA 3230
 17   CONTINUE                                                          ARA 3240
C*****END RADIATION PLUS CONVECTION AROUND ALL BOUNDARIES               ARA 3250
      RETURN                                                            ARA 3260
C*****BEGIN ERROR ANALYSIS                                              ARA 3270
 18   CALL MISQU(3,'ARATEF ERR1 - DISTANCE CHANGE VALUE = 0.0 - ARATE = ARA 3280
     1STEP,TIME,SECTIM=$',ITR,TIME,SECTIM)                              ARA 3281
      GO TO 20                                                          ARA 3290
 19   CALL MISQU(4,'ARATEF ERR2 - BOUNDARY NODE FACE IS NOT TYPE 3 - IFTARA 3300
     1YPE=,I,J,ISIDE=$',IFTYPE(ISIDE),I,J,ISIDE)                        ARA 3301
      GO TO 20                                                          ARA 3310
 20   RETURN                                                            ARA 3320
C*****END ERROR ANALYSIS                                                ARA 3330
      END                                                               ARA 3340
```

APPENDIX F. LISTING OF SKELETON SUBPROGRAMS FOR SAMPLE I/O.FUNCTIONs

```
      SUBROUTINE TEMFUN(NCALL,EXPNAM,NINTGR,INTGER)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION INTGER(2,500)
      INTEGER*2 INTGER
      COMMON /TEMPS/  TMPOLD(10000),TMPNEW(10000)
C*****BEGIN READING TEMP.S
      LUNIT=INTGER(1,1)
      REWIND LUNIT
      READ (LUNIT) IMAX,JMAX,KMAX,(TMPNEW(I),I=1,KMAX)
      DO 1 I=1,KMAX
 1    TMPOLD(I)=TMPNEW(I)
      REWIND LUNIT
      PRINT 1001,LUNIT,IMAX,JMAX,KMAX
 1001 FORMAT(1X,'END  TEMPERATURE  RESTORE  FROM  LUNIT',I4,' - IMAX,JMA
     1X,KMAX =',3I6)
C*****END READING TEMP.S
      RETURN
      END


      SUBROUTINE BOUFUN(NCALL,EXPNAM,NINTGR,INTGER)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION INTGER(2,500)
      INTEGER*2 INTGER
      PRINT 1001
 1001 FORMAT(///1X,'I/O SUBROUTINE BOUFUN CALLED'//)
      RETURN
      END


      SUBROUTINE MATFUN(NCALL,EXPNAM,NINTGR,INTGER)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION INTGER(2,500)
      INTEGER*2 INTGER
      PRINT 1001
 1001 FORMAT(///1X,'I/O SUBROUTINE MATFUN CALLED'//)
      RETURN
      END


      SUBROUTINE LISFUN(NCALL,EXPNAM,NINTGR,INTGER)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION INTGER(2,500)
      INTEGER*2 INTGER
      PRINT 1001
 1001 FORMAT(///1X,'I/O SUBROUTINE LISFUN CALLED'//)
      RETURN
      END
```

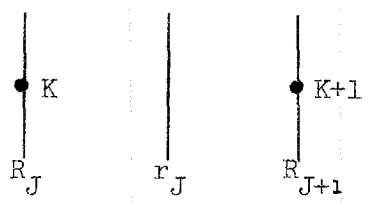
APPENDIX G. DERIVATION OF DIFFERENCE EQUATION FOR $\partial T/\partial n$

The normal derivative of the temperature is written in the form:

$$\partial T/\partial n_{K,S} = (\tau_{K,S} - T_K)/B_{K,S} \tag{2-15}$$

where the index K designates the nodal volume in which a heat balance is being computed, and the index S indicates which face of the nodal volume is being considered. $B_{K,S}$, then, is in the form of a distance and $\tau_{K,S}$ is in the form of a temperature at the neighboring node. There are four cases to consider:*

1. Neighboring node in same material



$$\frac{\partial T}{\partial n} = \left.\frac{\partial T}{\partial r}\right|_{r_J} \simeq \frac{T_{K+1} - T_K}{R_{J+1} - R_J} \tag{G-1}$$

so that $\tau_{K,S} = T_{K+1}$ and $B_{K,S} = R_{J+1} - R_J$.

2. Neighboring node in a different material

Let $\overline{T}$ indicate the temperature at r_J, the interface between the nodal volumes. Then continuity of heat flow at r_J requires:

$$- k_K \left.\frac{\partial T}{\partial r}\right|_{r_J^-} = - k_{K+1} \left.\frac{\partial T}{\partial r}\right|_{r_J^+} \tag{G-2}$$

*The derivations here are for the radial direction in r-z geometry.

The symbol r_J^- indicates that the derivative is to be evaluated just to the left of the interface, and r_J^+ indicates that the derivative is to be evaluated just to the right of the interface. Expanding T in a Taylor's series about r_J yields:

$$T_K = \overline{T} - (r_J - R_J) \left.\frac{\partial T}{\partial r}\right|_{r_J^-} \tag{G-3}$$

$$T_{K+1} = \overline{T} + (R_{J+1} - r_J) \left.\frac{\partial T}{\partial r}\right|_{r_J^+} \tag{G-4}$$

Substituting G-2 into G-4 and eliminating $\overline{T}$ we have

$$T_{K+1} - T_K = (R_{J+1} - r_J) \frac{k_K}{k_{K+1}} \left.\frac{\partial T}{\partial r}\right|_{r_J^-} + (r_J - R_J) \left.\frac{\partial T}{\partial r}\right|_{r_J^-} \tag{G-5}$$

so that

$$T_{K,S} = T_{K+1} \tag{G-6}$$

$$B_{K,S} = (R_{J+1} - r_J) \frac{k_K}{k_{K+1}} + (r_J - R_J) \tag{G-7}$$

3. Boundary node, mixed boundary condition

With the boundary condition:

$$\left.\frac{\partial T}{\partial r}\right|_{r_J} = a\,[T_{ext} - T(r_J)] \tag{G-8}$$

we have

$$T(r_J) = T_{ext} - \frac{1}{a} \left.\frac{\partial T}{\partial r}\right|_{r_J} \tag{G-9}$$

and expanding T in a Taylor's series, we have

$$T(R_J) = T(r_J) - (r_J - R_J) \left.\frac{\partial T}{\partial r}\right|_{r_J} ; \tag{G-10}$$

thus

$$T(R_J) = T_{ext} - (r_J - R_J + 1/a) \left.\frac{\partial T}{\partial r}\right|_{r_J} \tag{G-11}$$

so that

$$\tau_{K,S} = T_{ext} \tag{G-12}$$

$$B_{K,S} = r_J - R_J + 1/a$$

4. Boundary node, fixed temperature

With the boundary condition:

$$T(r_J) = T_b \tag{G-14}$$

substitution into G-10 yields

$$T(R_J) = T_b - (r_J - R_J) \left.\frac{\partial T}{\partial r}\right|_{r_J} \tag{G-15}$$

so that

$$\tau_{K,S} = T_b \tag{G-16}$$

$$B_{K,S} = r_J - R_J \tag{G-17}$$

APPENDIX H. STUDY OF STEADY-STATE CONVERGENCE

There are two aspects of convergence which we wish to discuss in this appendix. The first is the convergence of the iterative procedure for the solution of the finite difference equations (Section 2.3), whereas the second is the convergence of the solution of the finite difference equations to the solution of the differential equation as the mesh size becomes arbitrarily small.

To study these aspects of convergence we considered a fairly simple model problem, a one-region cylinder with a uniform heat source and with insulated top and bottom boundaries. The analytical solution of this one-dimensional problem can easily be obtained either for the linear case (constant thermal conductivity) or for selected nonlinear cases (e.g., a thermal conductivity linear in temperature). We used boundary condition (2-1) at the outer radius of the cylinder:

$$\left.\frac{dT}{dr}\right|_{r=b} = a[T_{ext} - T(b)] \quad . \tag{H-1}$$

In the case in which the thermal conductivity is a linear function of temperature we write:

$$k(T) = k(T_o)[1 + \alpha(T - T_o)] \quad . \tag{H-2}$$

We define a new dependent variable, $U(r)$, by means of a Kirchoff transformation: (see, for example, M. N. Ozisik, Boundary Value Problems of Heat Conduction, International Textbook Co., Scranton, Pa., 1968, p. 353)

$$U(r) = \int_0^{T(r)} k(\tau)\, d\tau \tag{H-3}$$

or, using (H-2)

$$U(r) = k_o\, T(r) \left[1 - \alpha\, T_o + \frac{\alpha}{2}\, T(r)\right] \tag{H-4}$$

with $k_o = k(T_o)$.

The differential equation (2-10) becomes

$$\frac{d}{dr}\left(k\,\frac{dT}{dr}\right) + \frac{k}{r}\,\frac{dT}{dr} + q = 0 \quad . \tag{H-5}$$

In terms of $U(r)$, however, we have the linear equation

$$\frac{d^2U}{dr^2} + \frac{1}{r}\,\frac{dU}{dr} + q = 0 \quad . \tag{H-6}$$

Since $\left.\frac{dT}{dr}\right)_{r=o} = 0$, the solution to (H-6) can be written in the form

$$\frac{dU}{dr} = -\,qr/2$$

$$U(r) = U(b) + q\,(b^2 - r^2)/4 \quad . \tag{H-7}$$

Taking the derivative of (H-3) we obtain

$$\frac{dU}{dr} = k(T)\,\frac{dT}{dr} \tag{H-8}$$

so that, at $r = b$,

$$k[T(b)]\,\left.\frac{dT}{dr}\right)_b = -\,qb/2 \tag{H-9}$$

and substituting (H-1) and (H-2) into (H-9), we obtain a quadratic equation for $T(b)$:

$$[\alpha\, T(b) + 1 - \alpha\, T_o]\,[T(b) - T_{ext}] = qb/2k_o\, a \quad . \tag{H-10}$$

Solving for T(b) and choosing the positive root

$$T(b) = \frac{2[T_{ext}(1 - \alpha T_o) + qb/2k_o a]}{1 - \alpha T_o - \alpha T_{ext} + \sqrt{(1 - \alpha T_o - \alpha T_{ext})^2 + 2\alpha\, qb/k_o a}} \quad . \qquad \text{(H-11)}$$

From (H-4)

$$U(b) = k_o T(b) \left[1 - \alpha T_o + \frac{\alpha}{2} T(b)\right] \qquad \text{(H-12)}$$

so that U(r) can be calculated from (H-7). Finally from (H-4) we can calculate

$$T(r) = \frac{U(r)}{k_o} \cdot \frac{2}{1 - \alpha T_o + \sqrt{(1 - \alpha T_o)^2 + 2\alpha U(r)/k_o}} \quad . \qquad \text{(H-13)}$$

This result coincides with the solution of (H-5) in the case of constant conductivity when $\alpha = 0$, as we would expect.

The finite-difference equations for the temperatures are obtained in Section 2.3 by imposing a heat balance on each nodal volume. In the one-dimensional problem studied in this appendix, the finite-difference equations can be solved explicitly for the temperatures in the constant-conductivity case. When $T_{ext} = 0$ we can write the temperature of node i as

$$\bar{T}_i = \frac{qb^2}{2K}\left[\frac{1}{ab} + \frac{1}{2}\left(1 - \frac{\sqrt{i}\,\sqrt{i-1}}{N}\right)\right] , \quad i = 1,2,\ldots,N \quad . \qquad \text{(H-14)}$$

In the constant-conductivity case with $T_{ext} = 0$, the solution of the differential equation is

$$T(r) = \frac{qb^2}{2k}\left[\frac{1}{ab} + \frac{1}{2}\left(1 - \frac{r^2}{b^2}\right)\right] \qquad \text{(H-15)}$$

at the nodes ($r = R_i$) we have

$$T_i = \frac{qb^2}{2k}\left[\frac{1}{ab} + \frac{1}{2}\left(1 - \frac{2i - 1 + 2\sqrt{i}\,\sqrt{i-1}}{4N}\right)\right] \tag{H-16}$$

so that the error is

$$T_i - \overline{T}_i = \frac{qb^2}{4k}\;\frac{1 - 2(i - \sqrt{i}\,\sqrt{i-1})}{4N} \quad . \tag{H-17}$$

Expanding $\sqrt{i-1}$ we have

$$\sqrt{i-1} = \sqrt{i}\sqrt{1 - \frac{1}{i}} = \sqrt{i}\,\left(1 - \frac{1}{2i} - \frac{1}{8i^2} - \ldots\right)$$

so that

$$T_i - \overline{T}_i = -\frac{qb^2}{64\,Nk}\left(\frac{1}{i} + \frac{1}{2i^2} + \ldots\right) \tag{H-18}$$

which is the negative for all nodes and has its largest absolute value for $i = 1$. The maximum absolute error then should appear at the first node, and should be inversely proportional to the number of mesh points and independent of the boundary constant a. Since the temperature at the first node will be smaller for large a (H-16), the relative error will be larger for large a. These features are all observed in the results of calculations with ORTHIS.

The rate of convergence of the iterative procedure used in ORTHIS has been shown by Stewart and Lick* to depend on the boundary constant a. They observed that the larger the value of a, the more rapid the convergence

---

*G. W. Stewart, III, and D. W. Lick, "Numerical Solution of a Thin Plate Heat Transfer Problem," Comm. of the ACM 11, 639 (1968).

should be. Since our model problem was different in two important respects (nonuniform mesh spacing and temperature-dependent conductivity), we ran several test cases with ORTHIS to investigate the effect on convergence of various values of β, the overrelaxation factor (Eq. 2-18), and of α, the slope of the conductivity function (H-2). Typical results are shown in Fig. H-1, where we have plotted the error measure

$$e_n = \sum_{K=1}^{KMAX} \left| \frac{T_K^{n-1}}{T_K^n} - 1 \right| \qquad \text{(H-19)}$$

as a function of n, the iteration number. In the region where the plotted curve becomes a straight line we can define N_{10}, the number of iterations required to reduce the error measure by a factor of 10. Results of several calculations are summarized in Table H-1.

Table H-1. Number of Iterations Required to Reduce Error by a Factor of 10

	a = .1				a = 2.0		
	$\alpha = 0$	$\alpha = .1$	$\alpha = .2$		$\alpha = 0$	$\alpha = .1$	$\alpha = .2$
β	N_{10}	N_{10}	N_{10}	β	N_{10}	N_{10}	N_{10}
1.85	167	131	120	1.40	59	58	57
1.75	304	240	221	1.30	75	73	72
1.65	456	359	332	1.20	93	91	90

These results are in qualitative agreement (for the $\alpha = 0$ cases) with those of Stewart and Lick.

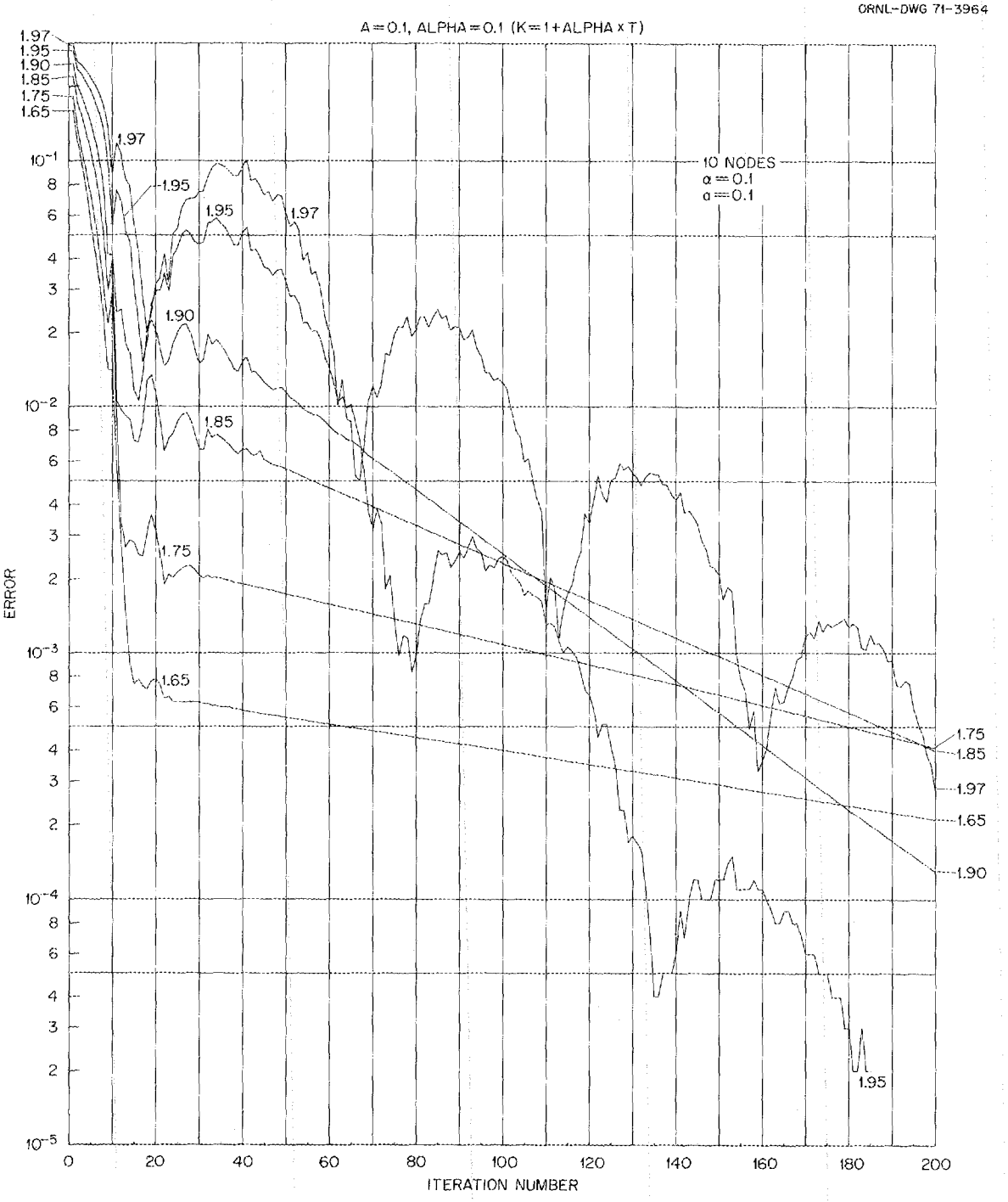


Figure H-1. Steady-State Convergence

APPENDIX I. CONTROL CARDS

The current versions of ORTHIS and ORTHAT are stored on the ZZZZZZ's disk at the computing center at Oak Ridge National Laboratory. The DSNAME for each code, respectively, is

DSNAME = AZ.G4.P31177.C10229.DURFEE.ORTHIS

DSNAME = AZ.G4.P31177.C10229.DURFEE.ORTHAT

The control cards which are necessary to use either of these codes are given below.

```
//aaaw JOB xxxxx,MSGLEVEL=1,CLASS=C
//STEP EXEC LINKNGO,REGION.GO=1024K                    (700K for ORTHIS)
//LINK.HEX DD DSN=AZ.G4.P31177.C10229.DURFEE.ORTHAT,      (or .ORTHIS)
//  VOL=SER=ZZZZZZ,UNIT=2314,DISP=SHR
//LINK.SYSIN DD *
    INCLUDE HEX
/*
//GO.FT50F001 DD *
    (Input Data Goes Here)
/*
//
```

The aaaw in columns 3-6 of the first card contain the job name, and the xxxxx in columns 12-16 contain the charge number. If a user wishes to generate plots with the steady-state code (ORTHIS), an additional DD card must be placed just before the //GO.FT50F001 card.

```
//GO.PLOTTAPE DD UNIT=TAPE7,VOL=SER=25,LABEL=(,NL),DISP=OLD
```

Likewise, if temperatures are to be saved on tape (see Section 3.12), then a DD card describing this tape must be placed just before the //GO.FT50F001 card. A typical card might look like:

```
//GO.FT02F001 DD UNIT=TAPE9,VOL=SER=2,DISP=OLD,LABEL=(,NL),
//  DCB=(RECFM=V,LRECL=3600,BLKSIZE=3604)
```

Of course, if any tapes are being used, they should be listed on the job submission form by their volume serial number (25 or 2 for DD cards above).

The job class must also be changed to F.

REFERENCES

1. R. S. Varga, Matrix Iterative Analysis, Prentice-Hall, Inc., Englewood, Cliffs, N. J., 1962.

2. D. R. Vondy, The Direct Criticality Search in Solution of Finite-Difference Eigenvalue Problems, ORNL-TM-2381 (March 1969), p. 45.

3. M. L. Tobias and T. B. Fowler, The Application of the Equipoise Iterative Method to Diffusion and Transport Theory, ANL-7050 (August 1965), p. 183.

4. K. D. Lathrop and F. W. Binkley, Theory and Use of the General-Geometry TWOTRAN Programs, LA-4432 (May 1970).

5. T. B. Fowler and D. R. Vondy, Nuclear Reactor Core Analysis Code: CITATION, ORNL-TM-2496 (Rev. 1) (January 1970); also M. L. Tobias, Mesh Spacing in Cylindrical Geometry Finite Difference Calculations in Diffusion Theory, ORNL-TM-957 (September 1964).

6. R. E. D. Stewart and F. C. Wessling, Jr., On Heat Conduction in Solids Having Temperature Dependent Properties, SC-RR-70-334 (June 1970).

7. R. C. Durfee, FORTRAN Dynamic Storage Allocation, ORNL-CF-70-8-59 (August 1970).

INTERNAL DISTRIBUTION

1. R. G. Alsmiller
2. D. E. Arnurius
3. S. J. Ball
4. M. Bender
5. C. E. Bettis
6. A. A. Brooks
7. W. E. Browning
8. V. R. Cain
9. T. D. Calton
10. D. W. Cardwell
11. R. S. Carlsmith
12. H. P. Carter
13. W. L. Carter
14. R. D. Cheverton
15. H. C. Claiborne
16. F. H. Clark
17. W. B. Cottrell
18. J. S. Crowell
19. W. Davis
20. J. G Delene
21. R. S. Dillon
22. L. Dresner
23-30. R. C. Durfee
31. J. R. Engel
32. C. H. Gabbard
33. R. B. Gallahar
34. W. R. Gambill
35. W. F. Gauster
36. N. B. Gove
37. G. E. Guest
38. M. Feliciano
39. M. H. Fontana
40. T. B. Fowler
41. A. P. Fraas
42. R. E. Funderlic
43. T. P. Hamrick
44. P. N. Haubenreich
45. R. W. Hendricks
46. E. C. Hise
47. H. W. Hoffman
48. F. J. Homan
49. W. H. Jordan
50. A. F. Joseph
51. L. Jung
52. F. B. Kam
53-87. P. R. Kasten
88. R. J. Kedl
89. J. J. Keyes
90. O. H. Klepper
91. B. B. Klima
92. T. S. Kress
93. M. E. Laverne
94. J. Lewin
95. G. H. Llewellyn
96. J. L. Lucius
97. R. N. Lyon
98. J. J. Manning
99. J. H. Marable
100. B. F. Maskewitz
101. R. D. McCulloch
102. H. A. McLain
103. S. E. Moore
104. F. R. Mynatt
105. F. H. Neill
106. D. B. Nelson
107. P. Nelson
108. C. W. Nestor
109. J. P. Nichols
110. C. E. Parker
111. L. F. Parsly
112. M. R. Patterson
113. S. K. Penny
114. R. B. Perez
115. A. M. Perry
116. H. P. Piper
117. B. E. Prince
118. R. P. Rannie
119. M. Reeves
120. M. W. Rosenthal
121. R. Salmon
122. J. P. Sanders
123. W. K. Sartory
124. L. B. Shappert
125. M. Simon-Tov
126. T. M. Sims
127. A. H. Snell
128. W. C. Stoddart
129. J. R. Tallackson
130. D. G. Thomas
131. M. L. Tobias
132. T. C. Tucker
133. W. D. Turner
134. D. R. Vondy
135. C. C. Webster

136. T. Welton
137. G. E. Whitesides
138. R. P. Wichner
139. R. E. Worsham
140. M. H. Yarosh
141-142. Central Research Library
143. Document Reference Section
144-148. Laboratory Records Department
149. Laboratory Records Department (RC)
150. ORNL Patent Office
151-152. Division of Technical Information Extension
153. Laboratory and University Division, ORO
154-203. Mathematics Division

EXTERNAL DISTRIBUTION

204. Prof. F. P. Brooks, Jr., University of North Carolina, Chapel Hill, North Carolina
205. W. R. Cowell, Argonne National Laboratory, Argonne, Illinois
206. R. H. Odegaarden, Fuel Fabrication and Transportation Branch, Division of Materials Licensing, U. S. Atomic Energy Commission, Washington, D. C. 20545
207. Prof. P. F. Pasqua, Dept. of Nuclear Engineering, University of Tennessee, Knoxville, Tennessee
208. C. V. L. Smith, U. S. Atomic Energy Commission, Washington, D. C.
209. R. A. Wagner, Idaho Nuclear Corporation, Idaho Falls, Idaho

ADDENDUM

BOUNDARY CONDITIONS FOR TRANSIENT HEAT TRANSFER BY RADIATION, FORCED CONVECTION, AND NATURAL CONVECTION

The heat flow per unit area per unit time at the outer boundary is assumed to be of the form

$$- k \left.\frac{\partial T}{\partial r}\right)^{\Delta}_{r_{JMAX}} = \epsilon\sigma(T_b^{\Delta 4} - T_{ext}^4) + h_c(T_b^{\Delta} - T_{ext}) + h_n\left|\frac{T_b^{\Delta} - T_{ext}}{T_c}\right|^{h_e}(T_b^{\Delta} - T_{ext}) \tag{1}$$

where the superscript Δ indicates that the boundary temperature T_b and the derivative $\frac{\partial T}{\partial r}$ are to be evaluated at $t + \Delta t$; T_{ext} is the external temperature and T_c is a temperature difference of $1^{\circ}F$; k is the conductivity of the surface material; ϵ is the emissivity of the surface material; σ is the Stefan-Boltzmann constant; h_c is the heat transfer coefficient for forced convection; and the heat transfer coefficient for natural convection is assumed to be of the form

$$h = h_n\left|\frac{T_b^{\Delta} - T_{ext}}{T_c}\right|^{h_e} . \tag{2}$$

We expand the temperature near the boundary in a Taylor series:

$$T_b^{\Delta} = T_{JMAX}^{\Delta} + (r_{JMAX} - R_{JMAX}) \left.\frac{\partial T}{\partial r}\right)^{\Delta}_{r_{JMAX}} \tag{3}$$

and substituting (1) into (3) we have

$$T_b^\Delta = T_{JMAX}^\Delta + (r_{JMAX} - R_{JMAX}) \left\{ - \frac{\varepsilon\sigma}{k} (T_b^{\Delta 4} - T_{ext}^4) - \left[\frac{h_c}{k} + \frac{h_n}{k} \left| \frac{T_b^\Delta - T_{ext}}{T_c} \right|^{h_e} \right] (T_b^\Delta - T_{ext}) \right\} \tag{4}$$

which can be written concisely as

$$F(T_b^\Delta) = 0 \ . \tag{5}$$

Equation (5) can be solved iteratively for the new boundary temperature by the Newton-Raphson method:

$$T_{b,i+1}^\Delta = T_{b,i}^\Delta - \frac{F(T_{b,i}^\Delta)}{F'(T_{b,i}^\Delta)} \ . \tag{6}$$

Our boundary condition is (cf Eq. 2-1)

$$T_b + \frac{1}{a} \left.\frac{\partial T}{\partial r}\right)_b = T_{ext} \quad \text{or} \quad - \left.\frac{\partial T}{\partial r}\right)_b = a(T_b - T_{ext}) \ . \tag{7}$$

We therefore define a total gradient constant a_T as

$$a_T = \frac{\varepsilon\sigma}{k} \frac{T_b^{\Delta 4} - T_{ext}^4}{T_b^\Delta - T_{ext}} + a_c + a_n \left| \frac{T_b^\Delta - T_{ext}}{T_c} \right|^{h_e} \tag{8}$$

or, writing the first quotient in factored form,

$$a_T = \frac{\varepsilon\sigma}{k} (T_b^\Delta + T_{ext})(T_b^{\Delta 2} + T_{ext}^2) + a_c + a_n \left| \frac{T_b^\Delta - T_{ext}}{T_c} \right|^{h_e} \tag{9}$$

where

$a_c = h_c/k$ and

$a_n = h_n/k$.

We have implemented these equations in the transient code (ORTHAT) for all outside surfaces by using the subroutine ARATEF (see Appendix E). This subroutine allows the gradient constant, a_T, to change at specified time steps. The user can write, for example

DIST CHANGE = 0.6 F. FOR TIM.STEPS 30-60 .

in the ***BOUNDARY*** section of the input. Subroutine ARATEF would then be called for time steps 30-60 to calculate a gradient constant a_T with additional input parameters given in Table 6.1.

Table 6.1

Parameter	Type of Heat Transfer	English Units	Default Value
ϵ	Radiation	none	0.9
h_n	Natural convection	BTU/hr ft^2 ^{o}F	0.19
h_e	Natural convection	none	1/3

These parameters are specified in DATA statements in the subroutine ARATEF. The reciprocal of the gradient constant for forced convection, $1/a_c$, will be given the value on the DIST CHANGE card.

To delete radiation as a heat transfer process the user can set ϵ to zero in the appropriate data statement. Similarly, $1/a_c$ or h_n can be set to zero to ignore forced or natural convection. Note that a new gradient constant, a_T, will be calculated for <u>every</u> boundary node not set at a fixed temperature (EDGE TEMP = ...) or insulated. If this is undesirable, the subroutine can be changed fairly easily.