ORNL/CSD/TM-15

HEATING5—An IBM 360 Heat Conduction Program

W. D. Turner
D. C. Elrod
I. I. Siman-Tov

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION FOR THE ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy $6.00; Microfiche $3.00

ORNL/CSD/TM-15

Contract No. W-7405 eng 26

COMPUTER SCIENCES DIVISION

HEATING5 –
AN IBM 360 HEAT CONDUCTION PROGRAM

W. D. Turner
D. C. Elrod
I. I. Siman-Tov

Date Published: March 1977

Performed for ERDA and for the U. S. Nuclear Regulatory Commission, Office of Nuclear Reactor Regulation, under Interagency Agreement ERDA No. 40-545-75.

UNION CARBIDE CORPORATION, NUCLEAR DIVISION
operating the

Oak Ridge Gaseous Diffusion Plant
Oak Ridge Y-12 Plant
Oak Ridge National Laboratory
Paducah Gaseous Diffusion Plant

for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

CONTENTS

CONTENTS (Contd.)

LIST OF FIGURES

LIST OF FIGURES (Contd.)

LIST OF TABLES

HEATING5 -

AN IBM 360 HEAT CONDUCTION PROGRAM

W. D. Turner
D. C. Elrod
I. I. Siman-Tov

ABSTRACT

HEATING5, a modification of the generalized heat conduction code HEATING3, is designed to solve steady-state and/or transient heat conduction problems in one-, two-, or three-dimensional Cartesian or cylindrical coordinates or one-dimensional spherical coordinates. The thermal conductivity, density, and specific heat may be both spatially and temperature-dependent. The thermal conductivity may be anisotropic. Materials may undergo a change of phase. Heat generation rates may be dependent on time, temperature and position, and boundary temperatures may be time-dependent. The boundary conditions, which may be surface-to-boundary or surface-to-surface, may be fixed temperatures or any combination of prescribed heat flux, forced convection, natural convection, and radiation. The boundary condition parameters may be time- and/or temperature-dependent. The mesh spacing can be variable along each axis. The code is designed to allow a maximum of 100 regions, 50 materials, and 50 boundary conditions. The maximum number of lattice points can be easily adjusted to fit the problem and the computer storage requirements. The storage requirements on an IBM 360 machine range from approximately 250K bytes for one lattice point to 1256K bytes for 6000 lattice points.

The point successive overrelaxation iterative method and a modification of the "Aitken δ^2 extrapolation process" are used to solve the finite difference equations which approximate the partial differential equations for a steady-state problem.

The transient problem may be solved using any one of several finite difference schemes. These include an implicit technique which can range from Crank–Nicolson to the Classical Implicit Procedure, an explicit method which is stable for a time step of any size, and the Classical Explicit Procedure which involves the first forward time difference. The solution of the system of equations arising from the implicit technique is accomplished by point successive overrelaxation iteration, and includes procedures to estimate the optimum acceleration parameter. The time step size for implicit transient calculations may be varied as a function of the maximum temperature change at a node. Transient problems involving materials with change-of-phase capabilities cannot be solved using the implicit technique with this version of HEATING5.

1. INTRODUCTION

HEATING5 is the latest version of "The HEATING Program," where HEATING is an acronym for Heat Engineering and Transfer in Nine Geometries. HEATING was originally developed by Liguori and Stephenson (Ref. 1) from Fowler and Volk's generalized heat conduction code, GHT (Ref. 2). Other modifications to HEATING have been reported previously (Ref. 3 and 4).

HEATING5, a modified version of HEATING3 (Ref. 4), has been stored on disks at the Computing Centers at Oak Ridge National Laboratory (ORNL) and the Oak Ridge Gaseous Diffusion Plant (ORGDP) and is available to users in Oak Ridge. This report, which is designed as a user's manual, discusses the capabilities of HEATING5.

The major improvement in the code is the incorporation of an implicit scheme to solve transient problems. Of the three basic algorithms which are available in HEATING5 to solve transient problems, the implicit scheme is the recommended approach for most problems. This scheme is written generally to include the Crank–Nicolson finite difference equations, the classical implicit or backwards Euler finite difference equations, or a linear combination of the two. The resulting system of equations is solved by the point successive overrelaxation iterative method, and the technique includes procedures to estimate the optimum acceleration parameter as a function of time. The time step size for the implicit transient calculations can be controlled explicitly through the input data or implicitly by specifying the maximum temperature change or maximum percent of relative change in temperature allowed at a node over a time step. The temperature-dependent parameters may be reevaluated as a function of the number of iterations for steady-state problems. Another modification allows selected materials to undergo a change of phase. However, the implicit technique for transient problems cannot be used for problems involving materials with change-of-phase capabilities with this version of HEATING5. Another feature which has been added to the code is the capability of solving one-dimensional, spherical models. Among other improvements incorporated in the new code are input and output modifications designed to facilitate data preparation and interpretation of results. The input formats for the material parameters, the heat generation rate functions and the initial temperature functions have been changed. The position-dependent functions and the time-dependent functions have been deleted. Analytical and tabular functions have been added to aid in the definition of input parameters. An option to allow the user to write his own subroutines to evaluate many of the input parameters has been added to the code. Thus, if an input parameter cannot be described with the built-in analytical or tabular functions, then the user may easily supply his own algorithm to evaluate the parameter. This concept is referred to as user-supplied subroutines. The boundary condition parameters may be time- and/or temperature-dependent or if they are defined in user-supplied subroutines, they can also be position-dependent. The thermal conductivity, density and heat capacity can also be time-dependent if they are defined in user-supplied subroutines. For two- and three-dimensional problems, the temperatures in each plane are printed in the form of a map which depicts the material boundaries. This feature enables one to monitor the temperature distribution in a plane with minimal effort. A nodal map accompanies the first temperature map which allows one to readily locate a node and its temperature.

The numerical techniques for the steady-state and transient calculations are discussed in Section 2. The use of the code to solve physical problems is presented in Section 3. An outline of the input data is included in Section 3.8. An output description is presented in Section 4.

The appendices include information on the control cards necessary to use the code at ORNL and ORGDP, nomenclature, information on the use of user-supplied subroutines, and sample problems.

The user is cautioned that this code is not a black box which digests the input data and automatically yields the correct solution to the physical problem. Care must be exercised in correctly simulating the physical problem as well as in interpreting the results from the code.

For steady-state problems, one must experiment with the mesh spacing in order to gain confidence in the numerical solution to a model of a physical problem. For instance, numerical solutions must be obtained for several different mesh spacings, then these solutions must be compared and the differences that are noted at points of interest must be acceptable. In addition, one must also experiment with the convergence criterion. When the criterion is satisfied, it only guarantees that the temperatures did not change more than a specified amount over the previous iteration. This is sufficient for many problems, but it is possible to have a problem which is converging so slowly that the convergence criterion is satisfied even though the last iterate is a very poor estimate of the true solution to the model. Again, one must make several calculations with different convergence criteria and compare the results before obtaining confidence in the solution. For transient problems, one must experiment with the size of the time increment as well as study the effects of varying the mesh spacing and convergence criteria.

2. NUMERICAL TECHNIQUE

2.1. Statement of the Problem

The HEATING5 Program solves the steady-state or transient heat conduction problem in either one, two, or three dimensions for either Cartesian or cylindrical coordinates or one dimension (radial) for spherical coordinates. For illustrative purposes, the equations and the discussion which follow are written for a three-dimensional problem in Cartesian coordinates.

First, the physical problem is approximated by a system of nodes each associated with a small volume. In order to define the nodes, a system of orthogonal planes is superimposed on the problem. The planes may be unequally spaced, but they must extend to the outer boundaries of the problem. A typical, internal node, which is defined by the intersection of any three planes is depicted in Fig. 2.1. Heat may flow from a node to each adjacent node along paths which are parallel to each axis. Thus for a three-dimensional problem, heat may flow from an internal node to each of its six neighboring nodes. The system of equations describing the temperature distribution is derived by performing a heat balance about each node.

The finite difference heat balance equation for node o in Fig. 2.1 is

$$C_o \frac{T_o^{n+1} - T_o^n}{\Delta t} = P_o^n + \sum_{m=1}^{6} {}_oK_m(T_m^n - T_o^n) \quad , \tag{2.1}$$

where T_m^n is the temperature of the mth node adjacent to node o at time t_n, ${}_oK_m$ is the conductance between nodes o and m, C_o is the heat capacitance of the material associated with node o, and P_o^n is the heat generation rate in the latter material at time t_n. Since planes go through the nodes and the material is homogeneous between any two successive planes along any axis, a node may be composed of as many as eight different materials, and the heat flow path between adjacent nodes may be composed of as many as four different materials positioned in parallel. For a three-dimensional problem, one C, one P, and six K's will be associated with each internal node at a particular time, t_n. These parameters are calculated as follows for node o:

$$C_o = \sum_{l=1}^{8} C_{pl}\rho_l V_l \quad , \tag{2.2}$$

$$P_o^n = \sum_{l=1}^{8} Q_l^n V_l \quad , \tag{2.3}$$

$${}_oK_m = \frac{1}{L_m} \sum_{\gamma=1}^{4} k_{m\gamma} A_{m\gamma} \quad , \tag{2.4}$$

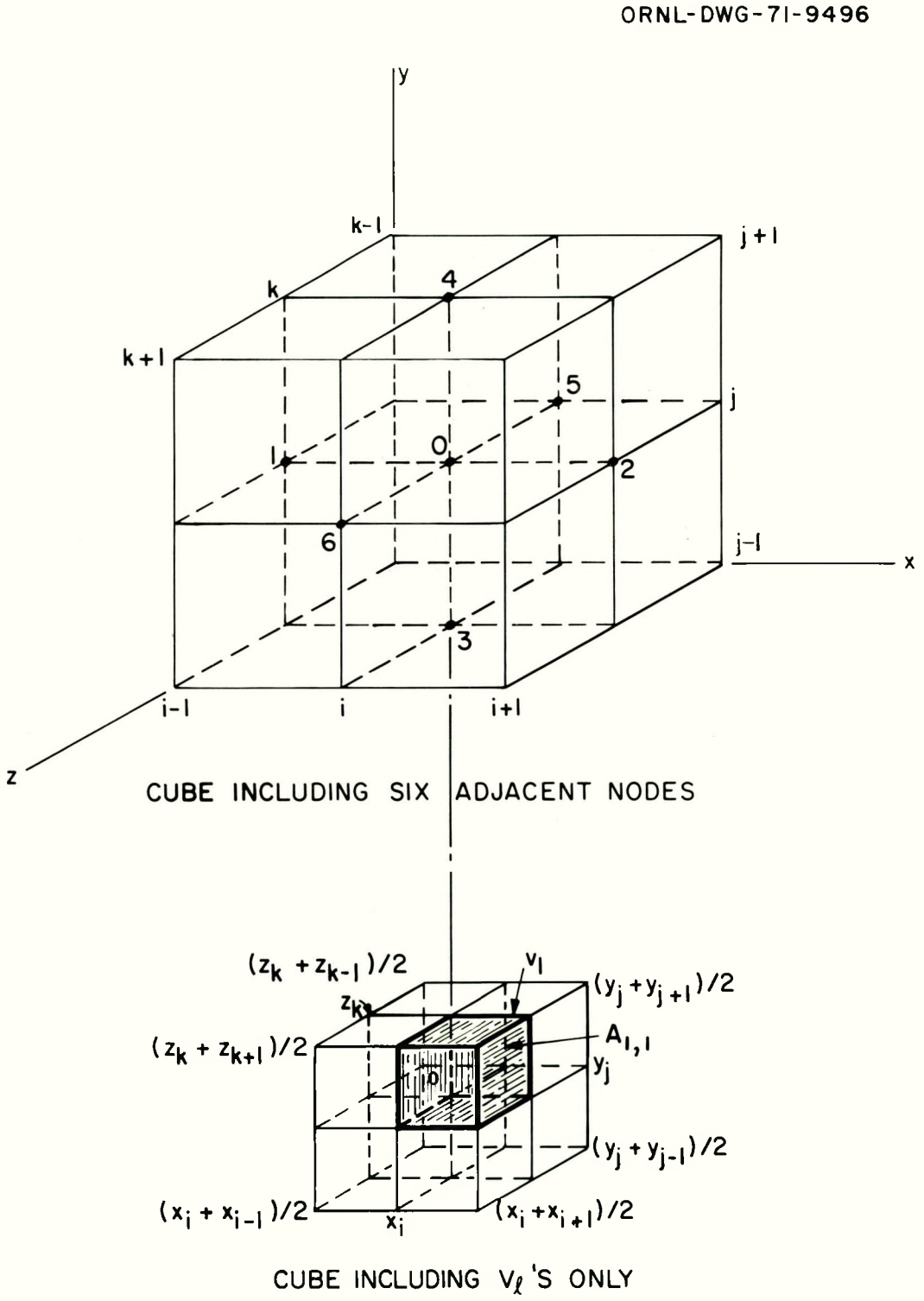

Fig. 2.1. HEATING5 Nodal Description for Three-Dimensional Problem

where

C_{pl} = specific heat for region l,

ρ_l = density for region l,

V_l = volume of region l,

Q_l^n = heat generation rate per unit volume in region l at time t_n,

L_m = distance between node o and adjacent node m,

$k_{m\gamma}$ = thermal conductivity for region γ between nodes o and m,

$A_{m\gamma}$ = cross-sectional area, normal to heat flow path, of region γ between nodes o and m.

With reference to Fig. 2.1, the V_l's and $A_{m\gamma}$'s are further defined, by using examples, as follows:

$$V_1 = \frac{(x_{i+1} - x_i)}{2} \quad \frac{(y_{j+1} - y_j)}{2} \quad \frac{(z_{k+1} - z_k)}{2} \quad , \tag{2.5}$$

$$A_{1,1} = \frac{(y_{j+1} - y_j)}{2} \quad \frac{(z_{k+1} - z_k)}{2} \quad . \tag{2.6}$$

Since nodes lying on a surface or nodes from one- or two-dimensional problems will not necessarily have six neighbors, the general heat balance equation for node i having M_i neighbors can be written as

$$C_i \frac{T_i^{n+1} - T_i^n}{\Delta t} = P_i^n + \sum_{m=1}^{M_i} {}_iK_{\alpha_m} (T_{\alpha_m}^n - T_i^n) \quad , \tag{2.7}$$

where α_m is the mth neighbor of the ith node. By choosing the increments between lattice lines small enough, the solution to the system of equations yields a practical approximation to the appropriate differential equation.

2.2. Steady-State Heat Conduction

For a steady-state heat conduction problem, the heat balance equation reduces to

$$P_i + \sum_{m=1}^{M_i} {}_iK_{\alpha_m} (T_{\alpha_m} - T_i) = 0 \tag{2.8}$$

since the left-hand side of Eq. (2.7) is zero.

If there are I nodal points, then since Eq. (2.8) will be applied at each node, there will be a system of I equations with I unknowns. The iterative technique which is used by HEATING5 to solve the system of equations is outlined below. First, solve Eq. (2.8) for T_i.

$$T_i = \frac{P_i + \sum_{m=1}^{M_i} {}_iK_{\alpha_m} \, T_{\alpha_m}}{\sum_{m=1}^{M_i} {}_iK_{\alpha_m}} \quad . \tag{2.9}$$

Since the values of T_{α_m} are unknown, the temperature at node i cannot be calculated directly from Eq. (2.9). However, an iterative procedure based on Eq. (2.9) can be used to estimate the steady-state temperature distribution. If an estimate to the temperature distribution exists, then Eq. (2.9) can be applied at each node, and hopefully, a better estimate to the temperature distribution will be obtained. Then, this new estimate can be used in Eq. (2.9) to produce an even better estimate. This iterative process can be written as

$$T_i^{(n+1)} = \frac{P_i + \sum_{m=1}^{M_i} {}_iK_{\alpha_m} \, T_{\alpha_m}^{(n)}}{\sum_{m=1}^{M_i} {}_iK_{\alpha_m}} \quad , \tag{2.10}$$

where the superscript (n) implies the nth iterate. Instead of using the results of Eq. (2.10) as the (n+1)st iterate, assume that it only yields an estimate and denote it as $\hat{T}_i^{(n+1)}$. Then define the (n+1)st iterate to be

$$T_i^{(n+1)} = T_i^{(n)} + \beta\left[\hat{T}_i^{(n+1)} - T_i^{(n)}\right] \quad , \tag{2.11}$$

where the relaxation factor, β, is limited to $0 < \beta < 2$. Notice that when $\beta > 1$, the new iterate is changed more than Eq. (2.10) specifies, and thus, the iterate is overrelaxed. Likewise, when $\beta < 1$, the iterate is underrelaxed. If $\hat{T}_i^{(n+1)}$ is replaced by Eq. (2.10), then Eq. (2.11) can be written as

$$T_i^{(n+1)} = (1-\beta)T_i^{(n)} + \beta\left[\frac{P_i + \sum_{m=1}^{M_i} {}_iK_{\alpha_m} T_{\alpha_m}^{(n)}}{\sum_{m=1}^{M_i} {}_iK_{\alpha_m}}\right] \quad . \tag{2.12}$$

If the nodes are numbered along the x-axis from left to right, then along the y-axis from bottom to top and finally along the z-axis from the smallest plane to the largest, then the rate of convergence for the iterative procedure can be increased by using the most recent value of the temperature in Eq. (2.12). This algorithm can be written as

$$T_i^{(n+1)} = (1-\beta)T_i^{(n)} + \beta\left[\frac{P_i + \sum_{m=1}^{L_i} {}_iK_{\alpha_m} T_{\alpha_m}^{(n+1)} + \sum_{m=L_i+1}^{M_i} {}_iK_{\alpha_m} T_{\alpha_m}^{(n)}}{\sum_{m=1}^{M_i} {}_iK_{\alpha_m}}\right] \tag{2.13}$$

where L_i is defined so that $\alpha_{L_i} < i < \alpha_{L_i+1}$. Varga (Ref. 5) refers to this method as the point successive overrelaxation iterative method. To increase the rate of convergence, an exponential approximation for Eq. (2.13) is made based on the temperature change from one iteration to the next. The algorithm based on this approximation is used instead of Eq. (2.13) to calculate the new temperatures for nodes having relative temperature changes exceeding 10^{-3}. However, the algorithm is designed to bound the temperature change so that the new temperature cannot be more than two times the old temperature. This prohibits the technique from diverging due to a bad estimate of the initial temperature distribution. The exponential approximation reduces to Eq. (2.13) for small temperature changes. Successive iterations are carried out by HEATING5 until

$$\left|\frac{T_i^{(n)} - T_i^{(n+1)}}{T_i^{(n+1)}}\right|_{\max} \leqslant \epsilon\,, \tag{2.14}$$

where ϵ is the specified convergence criterion.

Another extrapolation procedure which is commonly used to increase the rate of convergence in an iterative solution to a system of equations is the "Aitken δ^2 extrapolation procedure." Briefly, if $T^{(n-1)}$, $T^{(n)}$, and $T^{(n+1)}$ are the temperatures at a certain point at the n–1st, nth and n+1st iterations, respectively, and if

$$| [T^{(n)} - T^{(n-1)}] | > | [T^{(n+1)} - T^{(n)}] | \tag{2.15}$$

and

$$[T^{(n+1)} - T^{(n)}] [T^{(n)} - T^{(n-1)}] > 0, \tag{2.16}$$

then, a better estimate of the temperature is

$$T_{new} = T^{(n+1)} + \frac{[T^{(n+1)} - T^{(n)}]^2}{[T^{(n)} - T^{(n-1)}] - [T^{(n+1)} - T^{(n)}]} \tag{2.17}$$

Actually, HEATING5 uses a modification of Aitken's δ^2 method by calculating an extrapolation factor, B, and approximating Eq. (2.17) at each node with

$$T_i^{(n+1)} = \tilde{T}_i^{(n+1)} + B\,[\tilde{T}_i^{(n+1)} - T_i^{(n)}] \quad . \tag{2.18}$$

where $\tilde{T}_i^{(n+1)}$ represents the n+1st iterate at node i before extrapolation.

A HEATING5 extrapolation cycle is defined as follows. The code completes 20 iterations and checks to see if the maximum of the absolute values of the relative temperature changes over an iteration has decreased monotonically over the last ten iterations. If not, the cycle starts over. If so, the code will extrapolate only if the relative change in extrapolation factors over two consecutive iterations is less than 5% and the maximum of the absolute values of the relative temperature changes decreases monotonically over the same two iterations. The extrapolation factor, B, which is the same for each node, is based on two maximum relative temperature changes; between the n-1st and the nth iterations and between the nth and the n+1st iterations.

The value of β in Eq. (2.13) which will produce the optimum convergence rate for all points is difficult to obtain analytically for simple geometries and is practically impossible to obtain for complex geometries. If an input value is not supplied for β, then HEATING5 uses the default value of 1.9. If the rate of convergence appears to be slow, then HEATING5 reduces β by 0.1. The code determines whether or not the rate of convergence is slow in the following manner. It was noted above that during an extrapolation cycle, the relative temperature change over an iteration is monitored over ten consecutive iterations. If the relative temperature changes do not decrease monotonically over these ten iterations, then the current relative temperature change is compared with the one arising ten iterations earlier. If the current relative temperature change is greater than two-thirds of the old one, then the SOR technique is converging slowly. This process may be repeated until $\beta = 1.0$. However, the code will not increase β.

2.3. Transient Heat Conduction

HEATING5 is designed to solve a transient problem by any one of several numerical schemes. The first is the Classical Explicit Procedure (CEP) which involves the first forward difference with respect to time and is thus stable only when the time step is smaller than the stability criterion. Levy's modification to the CEP is the second scheme, and it requires the temperature distribution at two times to calculate the temperatures at the new time level. The technique is stable for a time step of any size. The third procedure, which is written quite generally, actually contains several implicit techniques which are stable for a time step of any size. One can use the Crank–Nicolson heat balance equations, the Classical Implicit Procedure (CIP) or backwards Euler heat balance equations or a linear combination of the two. The resulting system of equations is solved by point successive overrelaxation iteration. Techniques have been included in the code to approximate the optimum acceleration parameter for problems involving constant thermal parameters as well as those whose effective thermal conductances and capacitances vary with time or temperature. The implicit procedure in HEATING5 has not been designed to solve problems involving materials which are allowed to undergo a change of phase.

The implicit procedure using the Crank-Nicolson heat balance equation is the recommended technique for solving transient problems. Levy's modification to the CEP can be a useful tool for obtaining the solution to problems. However, one must experiment with the time step size before accepting the resulting solution.

The stability criterion for the CEP is a function of a temperature-dependent heat generation rate or heat flux. This fact is not accounted for in HEATING5. Also, Levy's modification to the CEP is based on a constant heat generation rate or heat flux with respect to temperature. If one attempts to use one of the explicit transient algorithms along with a temperature-dependent source or heat flux, then the code will write out a warning message indicating that the time step allowed by HEATING5 may not yield a stable solution.

Equation (2.7) is the basic heat balance equation for transient problems. However, the right-hand side is modified for all but the CEP.

2.3.1 *Classical Explicit Procedure*

For a transient heat conduction problem, the heat balance equation, Eq. (2.7), can be solved for T_i^{n+1} to give

$$T_i^{n+1} = T_i^n + \frac{\Delta t}{C_i}\left[P_i^n + \sum_{m=1}^{M_i} {}_iK_{\alpha_m}\left(T_{\alpha_m}^n - T_i^n\right)\right] \quad . \tag{2.19}$$

Since Eq. (2.19) expresses the temperature of the ith node at the n+1st time level in terms of temperatures at the nth time level, it is an explicit technique, and the algorithm is known as the Classical Explicit Procedure (CEP) or the Forward Difference Equation. HEATING5 uses Eq. (2.19) to solve transient heat conduction problems. The numerical solution obtained by using this technique is stable, provided the time step satisfies the following inequality (Ref. 6).

$$\Delta t \leqslant \left(\frac{C_i}{\sum_{m=1}^{M_i} {}_iK_{\alpha_m}}\right) \text{ minimum for all nodes} \quad . \tag{2.20}$$

2.3.2 *Levy's Modification to the Classical Explicit Procedure*

The limitation on the size of the time step as indicated in Eq. (2.20) means, in many practical problems, a very high ratio of computer time to actual time. In some cases computation costs become so high that the use of the algorithm defined by Eq. (2.19) becomes impractical. Levy [Ref. 7] proposed a modified explicit method which is stable for any time step desired. This method has been incorporated as an option in HEATING5. The basic equation used is

$$T_i^{n+1} = T_i^n + \frac{1}{1+Z_i}\left\{\frac{\Delta t}{C_i}\left[\sum_{m=1}^{M_i} {}_iK_{\alpha_m}\left(T_{\alpha_m}^n - T_i^n\right) + P_i^n\right] + Z_i\left[T_i^n - T_i^{n-1}\right]\right\} \quad , \tag{2.21}$$

where

Z_i = a factor for node i which will insure a stable solution for any time step Δt.

If from Eq. (2.20), $(\Delta t_{max})_i$ is the maximum time step allowed at node i for a stable solution in the regular explicit method, then the factor Z_i in Eq. (2.21) is defined as

$$Z_i = \begin{cases} 0, \text{if } \dfrac{\Delta t}{(\Delta t_{max})_i} \leqslant 1 \\ 0.5\left[\dfrac{\Delta t}{(\Delta t_{max})_i} - 1\right], \text{if } \dfrac{\Delta t}{(\Delta t_{max})_i} > 1 \end{cases} \quad . \tag{2.22}$$

Levy (Ref. 7) says that the accuracy is good if Z_i is zero for somewhat over half of the nodes. Of course, one must experiment with the size of the time step in order to obtain an acceptable solution.

2.3.3 *Implicit Procedure*

a. Heat Balance Equation

If the right-hand side of Eq. (2.7) is evaluated at t_{n+1} instead of t_n, then the scheme is known as the Classical Implicit Procedure (CIP) or the backwards Euler procedure. If the right-hand side of Eq. (2.7) is evaluated at $t_{n+1/2}$, then the algorithm is known as the Crank-Nicolson (CN) procedure. A general algorithm which includes both the CN technique and the CIP is

$$C_i^{n+\Theta} \frac{T_i^{n+1} - T_i^n}{\Delta t} = P_i^{n+\Theta} + \Theta\left[\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}(T_{\alpha_m}^{n+1} - T_i^{n+1})\right] + (1-\Theta)\left[\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}(T_{\alpha_m}^{n} - T_i^{n})\right] \tag{2.23}$$

where $0 \leqslant \Theta \leqslant 1$ and where the superscript $n+\Theta$ implies that the parameter is evaluated at time $t_{n+\Theta}$. If $\Theta = 0.5$, then Eq. (2.23) becomes the CN technique and if $\Theta = 1.0$, the algorithm is the CIP. When Θ is less than 0.5, the technique is no longer stable for any time step. Notice that Eq. (2.23) reverts to Eq. (2.7) when $\Theta = 0$. This algorithm has been incorporated into HEATING5 for $0.5 \leqslant \Theta \leqslant 1.0$.

b. Numerical Technique

If there are I nodes in the problem and if the heat balance equation, Eq. (2.23), is written for each node, then there will be I equations and I unknowns, and the resulting system of equations can be solved iteratively. The procedure that is used in HEATING5 is outlined below. If Eq. (2.23) is rewritten so that the temperatures at t_{n+1} are on the left-hand side, then the equation becomes

$$-\Theta\left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\, T_{\alpha_m}^{n+1}\right)+\left[\frac{C_i^{n+\Theta}}{\Delta t}+\Theta\left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\right)\right]T_i^{n+1}=H_i \tag{2.24}$$

where

$$H_i=\frac{C_i^{n+\Theta}}{\Delta t}\,T_i^n+P_i^{n+\Theta}+(1-\Theta)\left[\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}(T_{\alpha_m}^n-T_i^n)\right] . \tag{2.25}$$

If we let

$$D_i=\frac{C_i^{n+\Theta}}{\Delta t}+\Theta\left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\right) \tag{2.26}$$

and if we delete the superscript, n+1, on the temperature, T, then Eq. (2.24) can be rewritten as

$$-\Theta\left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\, T_{\alpha_m}\right)+D_iT_i=H_i \tag{2.27}$$

where it is now understood that T_i represents the temperature of node i at the new time level. If Eq. (2.27) is solved for T_i, then

$$T_i=\frac{\Theta\left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\, T_{\alpha_m}\right)+H_i}{D_i} . \tag{2.28}$$

Since the values of T_{α_m} are unknown, one cannot directly solve for the temperature at node i. However, if an estimate of the temperature distribution at the new time level exists, then Eq. (2.28) can be solved at each node, and hopefully, one will have a better estimate for the temperature distribution. The procedure can be repeated using this better estimate. This process can be continued until the estimates have converged to the approximation of the temperature distribution at the new time level. This algorithm can be written as

$$T_i^{(n+1)} = \frac{\Theta \left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta} \; T_{\alpha_m}^{(n)} \right) + H_i}{D_i} \tag{2.29}$$

where the superscript (n) on T_i refers to the nth iterate of the temperature at node i at the new time level.

Instead of using Eq. (2.29) in the iterative process, the technique can be refined further. First, consider Eq. (2.29) as only an approximation to the (n+1)st iterate or

$$\hat{T}_i^{(n+1)} = \frac{\Theta \left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta} \; T_{\alpha_m}^{(n)} \right) + H_i}{D_i} \tag{2.30}$$

and then define the (n+1)st iterate as

$$T_i^{(n+1)} = T_i^{(n)} + \omega \left[\hat{T}_i^{(n+1)} - T_i^{(n)} \right] \tag{2.31}$$

or

$$T_i^{(n+1)} = (1-\omega) \; T_i^{(n)} + \omega \; \hat{T}_i^{(n+1)} \tag{2.32}$$

where $0 < \omega < 2$. Thus, the change in temperature based on the value at the last iteration is more ($\omega > 1$) or less ($\omega < 1$) than the calculated value. Usually, the iterative procedure converges faster when $\omega \neq 1$, and thus, ω is commonly referred to as the acceleration parameter. Combining Eqs. (2.30) and (2.32), one obtains

$$T_i^{(n+1)} = (1-\omega)\ T_i^{(n)} + \omega \left[\frac{\Theta \left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\ T_{\alpha_m}^{(n)} \right) + H_i}{D_i} \right] \quad . \tag{2.33}$$

If we always use the most recent iterates on the right-hand side of Eq. (2.33), then the algorithm reduces to

$$T_i^{(n+1)} = (1 - \omega)\ T_i^{(n)} + \omega \left\{ \frac{\Theta \left[\left(\sum_{m=1}^{L_i} {}_iK_{\alpha_m}^{n+\Theta}\ T_{\alpha_m}^{(n+1)} \right) + \left(\sum_{m=L_i+1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\ T_{\alpha_m}^{(n)} \right) \right] + H_i}{D_i} \right\} \tag{2.34}$$

where $\alpha_{L_i} < i < \alpha_{L_i+1}$. It was pointed out in the previous section that this method is referred to as the point successive overrelaxation iterative method. HEATING5 applies Eq. (2.34) to each node until the convergence criteria have been met.

The convergence criteria are derived as follows. When the nth iteration has been completed, substitute the nth iterates into Eq. (2.27) and denote the heat residual as

$$R_i^{(n)} = H_i + \Theta \left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\ T_{\alpha_m}^{(n)} \right) - D_i T_i^{(n)} \quad . \tag{2.35}$$

Now normalize the heat residual by dividing by the right-hand side of Eq. (2.24) or

$$\frac{R_i^{(n)}}{H_i} = \frac{H_i + \Theta \left(\sum_{m=1}^{M_i} {}_iK_{\alpha_m}^{n+\Theta}\ T_{\alpha_m}^{(n)} \right) - D_i\ T_i^{(n)}}{H_i} \quad . \tag{2.36}$$

HEATING5 uses two convergence criteria based on this normalized heat residual. They are

$$\left(\frac{R_i^{(n)}}{H_i} \right)_{\max} \leqslant \epsilon_1 \tag{2.37}$$

and

$$\frac{\left(\frac{R_i^{(n)}}{H_i}\right)_{max}}{\left(\frac{R_i^{(1)}}{H_i}\right)_{max}} \leqslant \epsilon_2 \quad . \tag{2.38}$$

The iterative procedure is started in the following manner. For the first time step, the starting estimate is equal to the initial temperature distribution. Thereafter, the starting estimate at t_{n+1} is determined by

$$\hat{T}_i^{n+1} = T_i^n + (T_i^n - T_i^{n-1})(\Delta t_{n+1}/\Delta t_n) \quad . \tag{2.39}$$

c. Temperature-Dependent Properties

For problems involving temperature-dependent thermal properties, the iterative procedure is as follows. Initially, the thermal parameters are evaluated at the initial temperatures. Then, the point successive overrelaxation iterative method as described above is applied to obtain the temperature distribution at the new time level. However, since none of the thermal parameters are updated during this procedure, the converged temperatures are only an estimate to the temperature distribution. Thus, the thermal properties are reevaluated and the entire procedure is repeated until the technique converges to the temperature distribution at the new time level. This process contains two levels of iteration. The inner loop is the basic iterative process in the point successive overrelaxation iterative method while the outer loop iterates on the thermal parameters. Let $\hat{T}_i^{n,m}$ denote the temperature of node i after the mth iteration on the outer loop at time t_n. Upon the completion of the mth outer loop, the temperature at which the thermal parameters are evaluated is calculated as

$$\hat{T}_i^{n+\Theta} = (1 - \Theta)T_i^n + \Theta\, \hat{T}_i^{n+1,m} \, . \tag{2.40}$$

The temperature distribution has converged at time t_n when the L_1 norm of the relativized temperature difference over successive iterations is less than the prescribed value or

$$\|\Delta T^{n,m}\|_1 \leqslant \epsilon_3 \tag{2.41}$$

where

$$\|\Delta T^{n,m}\|_1 = \frac{1}{n} \sum_{i=1}^{I} \left| \frac{T_i^{n,m} - T_i^{n,m-1}}{T_i^{n,m}} \right| \quad . \tag{2.42}$$

d. Estimation of the Optimum Acceleration Parameter

The rate of convergence of the point successive overrelaxation iterative method is strongly dependent on the value of the acceleration parameter ω. The optimum value of the parameter, denoted as ω_o, is usually not known prior to the solution to a problem, and it is a function of time for problems whose effective thermal conductances and capacitances vary with time or temperature. Several techniques have been developed to estimate ω_o for transient problems with constant thermal properties. The method developed by Carré (Ref. 8) has been incorporated into HEATING5. Briefly, this method consists of estimating ω_o based on the behavior of a norm of the residual vector during the iterative procedure. The estimates are computed as a function of the iteration number until the process converges to a best estimate to the optimum value. Thereafter, the code uses this converged value as the acceleration parameter.

It was observed that this process was not satisfactory for problems involving temperature- and time-dependent conductances and capacitances, so an empirical process was developed and added to HEATING5 to estimate ω_o. For the initial time step ω is equal to unity. The code will attempt to update ω every N_ω time steps relative to the last time ω was changed. The criteria which are applied to determine whether or not the current value of ω is a good estimate to ω_o are based on the number of iterations required for the inner iterative loop to converge on the first pass through the outer iterative loop at some time step. When the code determines that an attempt to update ω should be made after completion of a particular time step, then the number of iterations for this time step is compared to the number for the time step immediately following the last modification to ω. If the change in the number of iterations is equal to or exceeds the criterion I_ω (an input value), then ω is increased according to

$$\omega^{n+1} = \omega^n + 0.1\ (2.0 - \omega^n) \tag{2.43}$$

where the superscript n refers to the value of ω at time t_n. On each subsequent time step a new estimate is made for ω_o using an algorithm similar to Eq. (2.43). However, ω may be either increased or decreased according to Table 2.1. When the change in the number of iterations for two consecutive time steps is less than the criterion J_ω (an input value), the code assumes that it has a good estimate for ω_o and it uses this value for the subsequent time steps until it is time to attempt another ω update. At this time, the entire procedure is repeated.

Table 2.1. Logic to Determine Whether ω Should be Increased or Decreased

Last update resulted in ω being	Change in number of iterations compared to previous time step: Increase	Change in number of iterations compared to previous time step: Decrease
Increased	Decrease	Increase
Decreased	Increase	Decrease

2.3.4 *Variable Time Step Size*

The time step size can be varied in a number of ways during the implicit transient calculations. First it can be controlled explicitly by specifying it as a constant size during the entire calculation or during prescribed time intervals throughout the transient. The time step size can also be varied explicitly by multiplying it by some prescribed factor after each time step subject to maximum and minimum values which may or may not be specified.

Finally, it can be varied implicitly in two ways. One can specify the maximum temperature change and the maximum percent of relative change in temperature allowed at a node over a time step. The time step size is either decreased if one of the calculated maximum values exceeds its respective criterion or is increased if both of the calculated maximum values is less than their respective criteria. The size is increased or decreased in the following manner. If one of the calculated values exceeds its respective criterion, then

$$f_{\Delta t} = \min\left(0.95 * \frac{criterion}{calculated\ value}, \text{TSFACT}\right)$$

where TSFACT is an input value representing the factor which is multiplied by the old time step size to obtain the new time step size. The factor 0.95 is to insure the decrease is large enough since the maximum temperature change is not linear with the changing time step size. The new time step size is then determined as

$$\Delta t_{new} = \max\left(\Delta t_{min}, f_{\Delta t} * \Delta t_{old}\right)$$

where Δt_{min} is an input value representing the smallest time step size allowed and Δt_{old} is the current time step size. If one of the calculated values is less than its respective criterion, then

$$f_{\Delta t} = \max\left(0.95 * \frac{criterion}{calculated\ value}, 1.0\right)$$

where the factor 0.95 is to insure the increase is not too large since the maximum temperature change is not a linear function of the time step size. Then

$$f_{\Delta t} = \min(f_{\Delta t}, \text{TSFACT}, 2.0)$$

where TSFACT was described above. The new time step size is then calculated as

$$\Delta t_{new} = \min(\Delta t_{max}, f_{\Delta t} * \Delta t_{old})$$

where Δt_{max} is an input value representing the largest time step size allowed. If both options are specified, the code uses the smaller of the two new time steps.

If the new time step size is smaller than the old one, the code rejects the temperature distribution it has just calculated and returns to the old time level. It then calculates a new temperature distribution at the new time level using the smaller time step size. If the code reduces the time step size NREP (currently 10) times, it writes out a warning message, reduces

the time step size to the minimum value, recalculates the temperature distribution using the new time step size and moves ahead to the next time level.

If the new time step size is greater than or equal to the old one, the code accepts the temperature distribution it has just calculated and moves on to the next time level. If a time step size has been reduced, the code will not allow it to be increased again until NREDUC (currently equal to 5) time steps have lapsed.

When the time step size is varied implicitly, the time step size and the associated criteria can be greatly affected by discontinuities in the boundary conditions, time-dependent functions, temperature-dependent functions, and especially the initial conditions. The user must be selective in choosing input parameters for these cases. One approach would be to control the time step size explicitly for short periods of time following these discontinuities. One may also need to start out with a small time step size initially to smooth out the data.

As the transient calculations approach a printout time, the time step size is automatically reduced to allow the temperature distribution to be printed out at the exact specified time. The old time step size is saved so that calculations can resume using the old time step after the printout.

2.4. Temperature-Dependent Thermal Properties

HEATING5 allows the thermal conductivity, k, the specific heat, C_p, and the density, ρ, to vary with temperature. The code determines the conductivity of the material between two nodes, i and j, by evaluating the temperature-dependent conductivity at an average of the temperatures of the two nodes. This temperature is calculated as

$$\bar{T}^{(n)} = \frac{T_i^{(n-1)} + T_j^{(n-1)}}{2} \tag{2.44}$$

after completion of the nth iteration for steady-state problems, as

$$\bar{T}^{n} = \frac{T_i^{n-1} + T_j^{n-1}}{2} \tag{2.45}$$

after completion of the nth time step for transient problems involving one of the explicit techniques, and as

$$\bar{T}^{n+\Theta} = \frac{\hat{T}_i^{n+\Theta} + \hat{T}_j^{n+\Theta}}{2} \tag{2.46}$$

during the calculation of the temperature distribution at time t_{n+1} for transient problems involving the implicit procedure. The temperatures denoted as $\hat{T}_i^{n+\Theta}$ in Eq. (2.46) are evaluated according to Eq. (2.40). For transient problems, the specific heat and density are determined

for node i by evaluating the respective temperature-dependent function at T_i^n after completing the nth time step using one of the explicit techniques and at $\hat{T}_i^{n+\Theta}$ as determined by Eq. (2.40) during the calculation of the temperature distribution at time t_{n+1} using the implicit procedure. In addition, the thermal conductivity of a material can be anisotropic. See Section 3.6.7 for details of how to utilize this option.

2.5. Boundary Conditions

HEATING5 possesses a variety of boundary conditions to enable the user to model his physical problem as accurately as possible. In general, a boundary condition is applied along a surface of a region and heat is transferred from a surface node to a boundary node or to the corresponding node on the opposing parallel surface. Surface nodes are actually internal nodes which are located on the edge of a region. Boundary nodes are dummy nodes and their temperatures are not calculated by the code but are specified as input to the code. These temperatures are only used to calculate the heat flow across a boundary surface. The boundary conditions which can be applied over the surface of a region in the current version of HEATING5 are listed below.

a. The temperature on the surface of a region can be specified as a constant or a function of time.

b. The heat flux across the surface of a region can be specified directly as a constant or a function of time and/or surface temperature.

c. The heat flux across the surface of a region can be specified indirectly by defining the heat transfer mechanism to be forced convection, radiation and/or natural convection.

The numerical techniques used in calculating the temperatures of surface nodes associated with a boundary condition are discussed below. The temperatures of nodes on surfaces whose temperatures are specified are not calculated from Eq. (2.7). Instead, the surface node temperature is set equal to the specified value. When a heat flux is specified across a surface, then for each node on that surface, the specified heat flux is multiplied by its surface area normal to the heat flow path associated with the boundary condition, and the result is added to the heat generation term, P_i^n, in Eq. (2.7). If the heat flux is temperature-dependent, then it is evaluated at the average temperature of the related surface node and boundary node for surface-to-boundary boundary conditions and the average temperature of opposing surface-nodes for surface-to-surface boundary conditions. Simulation is not required for insulated boundaries. Heat is simply not allowed to cross the surface.

The boundary conditions are classed as either surface-to-boundary (type 1), isothermal (type 2), or surface-to-surface (type 3). Boundary conditions of the surface-to-boundary type are used to define heat transfer between a surface node and a boundary node. The temperature of the boundary node is specified and can be a function of time. Surface-to-surface boundary conditions are used to define heat transfer between parallel surfaces. In this case, heat is transferred between a node on one surface to the corresponding node on the opposing surface. In other words, in Fig. 2.2, surface-to-surface boundary conditions could be utilized to describe the heat transfer process between nodes 1 and 2, nodes 3 and 4, and nodes 5 and 6.

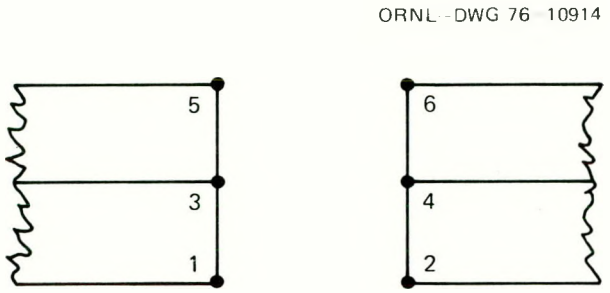

Fig. 2.2. Surface-to-Surface Heat Transfer

For both surface-to-boundary and surface-to-surface boundary conditions, the leakage term in Eq. (2.7) is calculated as follows:

$$\left[{}_iK_{\alpha_m}(T^n_{\alpha_m} - T^n_i)\right] = {}_iK_b(T^n_b - T^n_i) \tag{2.47}$$

where ${}_iK_b$ is the effective conductance from surface node i to boundary node b or the opposing surface node b, and T^n_b is either the temperature of boundary node b or the opposing surface node b at time t_n. The effective conductance is calculated as

$${}_iK_b = hA \tag{2.48}$$

where h is the effective heat transfer coefficient, and A is the surface area, normal to the heat flow path, of node i associated with the boundary condition.

The effective heat transfer coefficient is defined as

$$\begin{aligned} h = h_c + h_r \left[(T^n_i)^2 + (T^n_b)^2\right]\left[T^n_i + T^n_b\right] \\ + h_n \left|T^n_i - T^n_b\right|^{h_e} \end{aligned} \tag{2.49}$$

where

h_c is the heat transfer coefficient for forced convection,

h_r is a coefficient (product of the gray body shape factor and Stefan-Boltzman constant) for radiative heat transfer,

h_n and h_e are, respectively, the coefficient and exponent for the term simulating the effects of natural convective heat transfer or some other heat transfer process such as boiling.

The parameters h_c, h_r, h_n, and h_e must be specified by the user and can be time- and/or temperature-dependent. When h_c, h_r, h_n, and h_e are temperature-dependent, they are evaluated at the average temperature of the opposing surface-nodes for surface-to-surface boundary conditions. For surface-to-boundary boundary conditions, h_c, h_n, and h_e are evaluated at the average temperature of the related surface node and boundary node, whereas h_r is evaluated at the temperature of the surface node. For surface-to-boundary boundary conditions, the boundary temperature, T_b^n, must also be supplied by the user. The temperatures are entered in either °F or °C, and the code converts them to absolute degrees when a radiative boundary condition exists. In computing the effective conductance for a surface-to-surface boundary condition across a radial gap, the code uses the cross-sectional area at the smaller radius bounding the gap. This is important for the user to remember when he evaluates the radiation shape factor in h_r.

HEATING5 is designed so that, simultaneously, one may consider surface-to-surface heat transfer across a region as well as conduction through the region. This is accomplished by defining the region to contain a material as well as by defining surface-to-surface boundary conditions across parallel surfaces of the region. Also, one may consider surface-to-surface heat transfer across a gap as well as surface-to-boundary heat transfer along the edge of the gap. This is done by defining the gap as a gap region (i.e., it does not have a material associated with it) with surface-to-surface boundary conditions applied across parallel surfaces of the region. Then surface-to-boundary boundary conditions are defined on the adjacent material regions at the surfaces defining the edges of the gap. Note: A surface-to-boundary boundary condition can be applied along the surface of a region only if there is no region adjacent to it or the adjacent region is defined as a gap region. See Section 3.2 for an example.

2.6. Change of Phase

Selected materials are allowed to undergo a phase change during the transient calculations. The following technique is used in calculating the temperatures of nodes composed of materials which can have a change of phase. Let the melting ratio, $X1_i$, be the ratio of heat which has been absorbed after the transition temperature has been reached to the total heat needed to complete the phase change for a material in node i. Unless an input value is specified, the initial melting ratio is calculated as

$$X1_i = \begin{cases} 0.0, & T_i^\circ < T_{melt,i} \\ 1.0, & T_i^\circ \geqslant T_{melt,i} \end{cases} \tag{2.50}$$

where

T_i° is the initial temperature of node i and

$T_{melt,i}$ is the smallest phase-change or transition temperature associated with node i.

If the melting ratio of a node is zero, its temperature is allowed to increase until it reaches the transition temperature of the material associated with it. Then, the temperature of the node is held at the transition temperature, and the material is allowed to change phase in the following manner. The incremental melting ratio over the nth time step is calculated as

$$\Delta X1_i^n = \frac{\Delta q_i^n}{\rho_{i,m}^n \; H_m \; V_{i,m}} \tag{2.51}$$

where

Δq_i^n is the net heat into node i during the nth time step,

$\rho_{i,m}^n$ is the density of material m evaluated at T_i^n,

H_m is the latent heat of material m, and

$V_{i,m}$ is the volume of material m associated with node i.

This incremental ratio is added to the current value of $X1_i$ at each time step until $X1_i$ exceeds unity. Any excess heat remaining after a change of phase is used to change the temperature of the node as follows:

$$\Delta T_i^n = (X1_i^n - 1.0) \; \rho_{i,m}^n \; H_m \; V_{i,m} / C_i^n \; , \tag{2.52}$$

for $X1_i^n > 1.0$ where C_i^n is the heat capacitance of node i during the nth time step. Then, the melting ratio is set to unity, and the temperature of the node is allowed to increase. Likewise, if the melting ratio of a node is unity, its temperature is allowed to decrease until it reaches the transition temperature of the material associated with it. Then, the temperature of the node is held at the transition temperature, and the material is allowed to change phase in the following manner. The incremental melting ratio, Eq. (2.51), is added to the current value of $X1_i$ at each time step until $X1_i$ is less than zero. Any excess heat remaining after a change of phase is used to change the temperature of the node as follows:

$$\Delta T_i^n = X1_i^n \; \rho_{i,m}^n \; H_m \; V_{i,m} / C_i^n \tag{2.53}$$

for $X1_i^n < 0.0$. Then, the melting ratio is set to zero, and the temperature of the node is allowed to decrease.

If a node is associated with more than one material which can change phase, then each material is allowed to change phase independently. The materials are ordered by increasing transition temperature. If the temperature of the node is increasing, then its temperature is not allowed to exceed the lowest transition temperature until the melting ratio increases from zero to unity. Once the melting ratio reaches unity, it is fixed there, and the temperature of the node is allowed to increase until it reaches the second transition temperature. Then, the melting ratio is set equal to zero, and the temperature of the node is not allowed to increase until the melting ratio increases from zero to unity. Once the melting ratio reaches unity, the temperature of the node is allowed to increase again. This process is repeated until each material associated with the node changes phase. The logic is similar when the temperature of the node is decreasing. However, the melting ratio of a node is set to zero when its temperature is between transition temperatures of the materials associated with it.

It must be emphasized that this technique will allow changes of phase in both directions for as many times as it may be needed.

2.7. Initial Temperatures

Unless it is explicitly specified, the initial temperature at a node is calculated as a volume-average and a heat-content-average of the specified initial temperatures of the materials associated with the node for steady-state and transient cases, respectively.

3. INPUT DESCRIPTION

3.1. General

This section is designed to guide the user through the steps necessary to solve a heat transfer problem with HEATING5. First, a set of instructions is presented to aid the user in representing the geometrical configurations of the problem with a lattice of points. Then, the functions used by the code to define the material parameters are given. A detailed discussion of the input cards is presented followed by a brief outline of the input data for the benefit of the user who is familiar with the data preparation. In preparing the input data, any consistent set of units may be used in the HEATING5 program except for problems involving radiation. Then, all temperature units must be in either degrees Celsius or Fahrenheit. The units associated with the algorithms which appear in user-supplied subroutines must be consistent with those of the input data.

3.2. Regions

First, the configuration of the problem is approximated by dividing it into regions, depending on the shape, material structure, indentations, cutouts, and other deviations from the general geometry. In some cases, zoning into regions must be done in order to describe a specific boundary condition or a material whose thermal conductivity, density, or specific heat is a function of position. There are three basic rules governing region division:

(1) Boundary lines or planes must be parallel to the coordinate axes (two points, four lines, or six planes are required to enclose a region in one-, two-, or three-dimensional geometry, respectively).

(2) A region may contain at most one material (however, many regions may contain the same material). A gap region does not contain a material.

(3) When a boundary condition is defined along the boundary of a region, it must apply along the full length of the boundary line for two-dimensional problems and over all of the boundary plane for three-dimensional problems.

Consider, for example, a case consisting of a simple rectangle in x - y geometry, half of which contains one material, and the other half a second material, as depicted in Fig. 3.1.

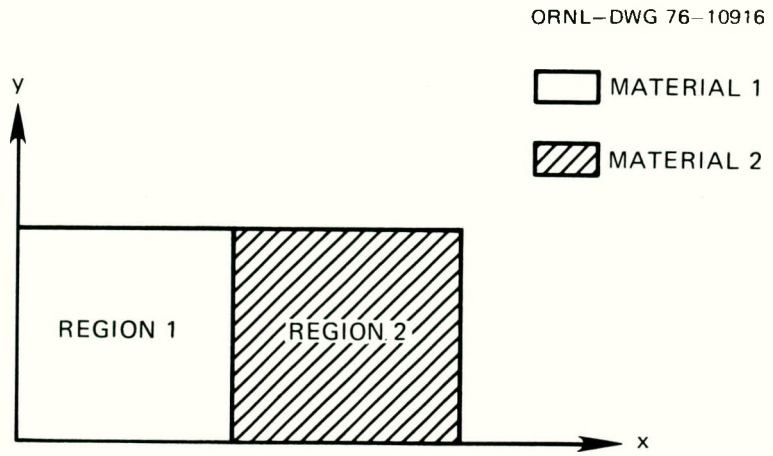

Fig. 3.1. HEATING5 Region Description for Two-Dimensional, Rectangular Model Composed of Two Materials.

This elementary case would require two regions (as indicated)—one for each material. If the upper right corner of the rectangle is omitted as in Fig. 3.2, three regions are required as shown.

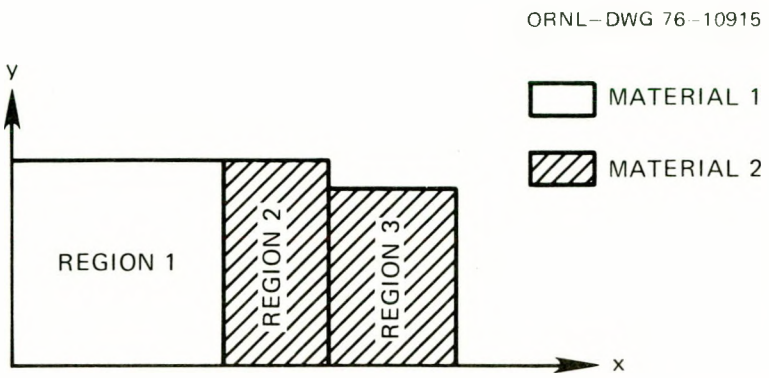

Fig. 3.2. HEATING5 Region Description for Two-Dimensional, Rectangular Model with Indentation.

The division of the right half of the rectangle into two regions accounts for the indented or cutout upper right corner. Note that regions 2 and 3 of Fig. 3.2 contain the same material.

Now consider the case of Fig. 3.2, introducing boundary conditions as in Fig. 3.3. The left boundary of the left-most rectangle now contains two different boundary types. Thus, in accordance with the third basic rule, region zoning is performed to account for the different boundary conditions, and an additional region is required.

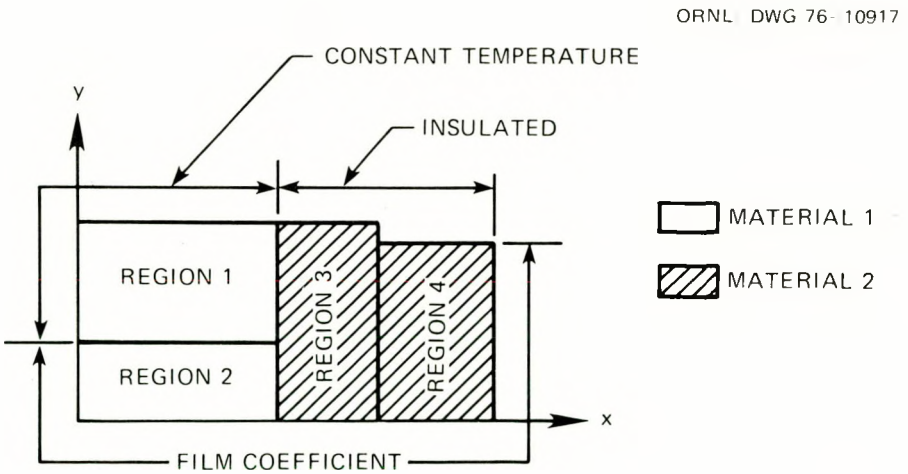

Fig. 3.3. HEATING5 Region Description for Two-Dimensional, Rectangular Model with Various Boundary Conditions.

To specify heat transfer between parallel surfaces, one defines a region whose boundaries include the two parallel surfaces. Then, the boundary condition describing the heat transfer process (type 3, see Section 3.6.10a) is applied along both of the surfaces of this region. Although the regions adjoining the parallel surfaces involving the surface-to-surface heat transfer may be composed of more than one material, they must be defined and must contain a material. The region itself may or may not contain a material. In Fig. 3.4, surface-to-surface heat transfer cannot be defined between the left and right boundaries of Region 3 since part of the area adjoining the right boundary is undefined.

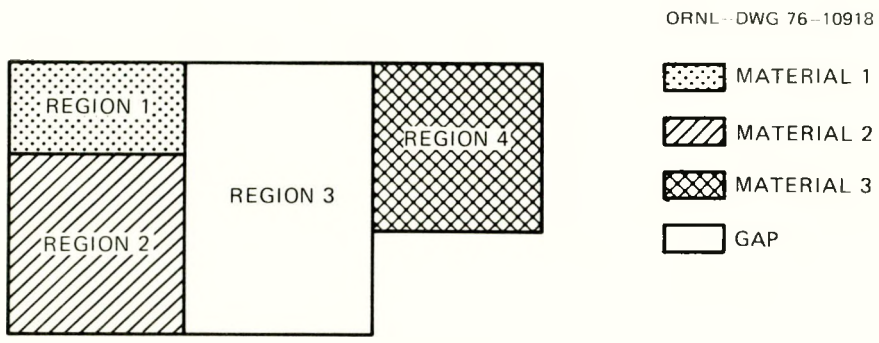

Fig. 3.4. Region Description for Two-Dimensional, Rectangular Model Involving Surface-to-Surface Boundary Conditions, Incompatible with HEATING5.

In Fig. 3.5 surface-to-surface boundary conditions can be applied along the left and right sides of Region 3.

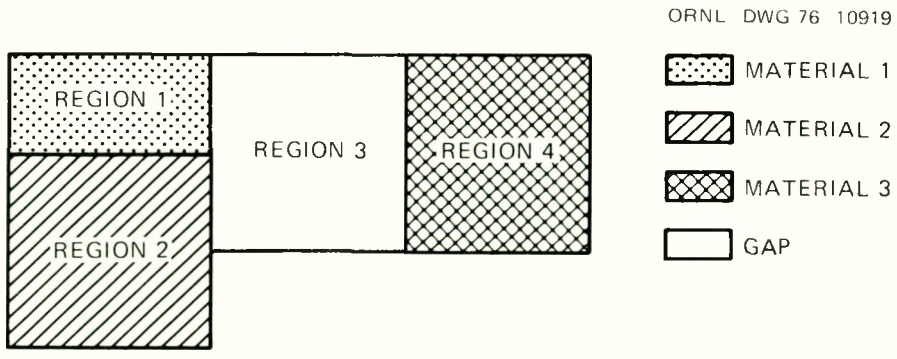

Fig. 3.5. HEATING5 Region Description for Two-Dimensional, Rectangular Model Involving Surface-to-Surface Boundary Conditions.

If a surface-to-surface (type 3) boundary condition has been defined along a surface of a region and a surface-to-boundary (type 1) boundary condition is desired along the same surface, then the type 1 boundary condition must be applied along the surface of the adjoining region. In Fig. 3.5, surface-to-surface boundary conditions can be applied along the left and right sides of Region 3 while a surface-to-boundary boundary condition can be applied along the left side of Region 4. This can be done only if Region 3 is a gap region. See Section 2.5 for more details.

3.3. Lattice Arrangement

The second requirement for describing the overall configuration is to construct a set of lattice lines perpendicular to each axis and extending the entire length of the remaining coordinates. The lattice lines are really points, lines, or planes for a one-, two-, or three-dimensional problem, respectively. However, the phrase, lattice line, will be used for illustrative purposes. The lattice is defined in the following manner. The lattice lines are divided into two classes: gross lattice lines and fine lattice lines. A gross lattice line must be specified at both region boundaries along each axis. Fine lattice lines, equally spaced, may appear between two consecutive gross lattice lines to create a finer mesh. If unequal mesh spacing is desired within a particular region, then gross lattice lines may appear within that region. A nodal point is defined by each lattice point in one-dimensional problems, by each intersection of two lattice lines appearing in a material region or on its boundary in two-dimensional problems, and by every intersection of three lattice planes appearing in a material region or on its boundary in three-dimensional problems. The points are numbered consecutively by the program at the intersection of each X- (or R-), Y- (or θ-), and Z-plane starting with the planes nearest the origin and changing the X- (or R-) plane most rapidly and the Z-plane least. Temperatures are calculated at each such nodal point.

3.4. Analytical and Tabular Functions

The analytical and tabular functions are built-in functions which may be used to aid in the description of some of the input parameters. An analytical function is defined by

$$F(v) = A_1+A_2v+A_3v^2+A_4\cos(A_5v) +A_6\exp(A_7v)+A_8\sin(A_9v)+A_{10}ln(A_{11}v) \quad . \tag{3.1}$$

A tabular function is defined by a set of ordered pairs, $(v_1, G(v_1))$, $(v_2, G(v_2))$,..., $(v_n, G(v_n))$, where the first element of a pair is the independent variable and the second is the corresponding value of the function. In order to evaluate the tabular function at some point, the program uses linear interpolation in the interval containing the point. The set of ordered pairs must be chosen so that the independent variable is arranged in ascending order or

$$v_1 < v_2 < v_3 < \ldots < v_{n-1} < v_n \; . \tag{3.2}$$

If the function must be evaluated at some point outside of the defined domain of the function, then

$$G(v) = \begin{cases} G(v_1), & v < v_1 \\ \\ G(v_n), & v > v_n \end{cases} \quad . \tag{3.3}$$

3.5. Input Parameters

The input parameters included in Table 3.1 must be defined by a function having the following form:

$$P(x,y,z,t,T) = P_0 \cdot f(x,y,z,t,T) , \tag{3.4}$$

where P_0 is a constant factor and f(x,y,z,t,T) may be a product of analytical and tabular functions, such as

$$f(x,y,z,t,T) = F_i(x)\cdot F_j(y)\cdot F_k(z)\cdot F_l(t)\cdot F_m(T) . \tag{3.5}$$

In observing the definitions of the input parameters (see Table 3.1) which may be defined by Eq. (3.4), it is noted that only the volumetric heat generation rate is a function of all the independent variables. Thus, if any variable is omitted from the definition of the parameter, then the corresponding factor is set equal to unity in Eq. (3.5). The constant factor, P_0, is part of the input data, and its value appears on the data card which is used to define the parameter. If one of the input parameters is to be a function of position, time, or temperature, then the appropriate index or indices, i, j, k, *l* or m, from Eq. (3.5) are entered on a data card, too. If an index is positive, then it defines the number of the analytical function for the respective variable. If it is negative, then the absolute value of the index defines the number of the tabular function for the respective variable. If any of the defined factors for a parameter are omitted from the input data, then that particular factor is set equal to unity in Eq. (3.5). If none of the factors are defined in the input, then that particular parameter is treated as a constant. If the value of P_0 is zero or is left blank on the data card and if the data indicates that the parameter is to be a function of position, time or temperature, then P_0 is set equal to unity in Eq. (3.4). If the value of P_0 is zero or is left blank and if the data indicates that the parameter is not a function of position, time, or temperature, then P_0 is set equal to zero in Eq. (3.4).

If the parameter cannot be defined by a product of analytical and tabular functions as indicated in Eq. (3.5), then the user may supply his own subroutine to evaluate Eq. (3.4). Table 3.1 contains each input parameter and the name of the corresponding subroutine which must be supplied if the user wishes to create his own function. Table 3.1 also includes the independent variables which may be used to define each parameter. For further details involving user-supplied subroutines, see Appendix C.

If the thermal conductivity of a material is anisotropic, then it must be defined in user-supplied subroutine CONDTN (see Section 3.6.7).

3.6. Card Description

A detailed description of the card input is presented below. Except for Card 1, the M cards, and the deck composed of the IT cards, the input data are arranged on each card in 9-column fields. All integers must be right-adjusted, i.e., the last digit of each integer must appear in a column which is a multiple of 9. Except for the IT cards, columns 73 through 80 of each card are reserved for identification to aid the user in the preparation and handling of the data.

3.6.1. *Card 1 - Title of Problem*

This card, which can contain alphanumeric characters in the first 72 columns, contains a descriptive title for the problem. The card itself cannot be omitted although it may be left blank.

Table 3.1. Dependence of Input Parameters

Parameter	Function of					Related User-Supplied Subroutine
	x	y	z	t	T	
k				(x)	x	CONDTN
ρ				(x)	x	DNSITY
C_p				(x)	x	CPHEAT
T_o	x	x	x	(x)		INITTP
Q	x	x	x	x	x	HEATGN
T_b				x		BNDTMP
h_c	(x)	(x)	(x)	x	x	CONVTN
h_r	(x)	(x)	(x)	x	x	RADITN
h_n	(x)	(x)	(x)	x	x	NATCON
h_e	(x)	(x)	(x)	x	x	NCONEX
h_f	(x)	(x)	(x)	x	x	BNFLUX

NOTE: The x enclosed in parentheses, (x), indicates that although the parameter cannot be an explicit function of the indicated variable using normal input data, this variable is initialized in the respective user-supplied subroutine. Thus, the parameter can be a function of the indicated variable if it is so defined in its respective user-supplied subroutine.

3.6.2. *Card 2 - Input Parameters*

All eight entries in this card are integers.

a. *Maximum CPU Time*

When one submits a job to the computer, the maximum CPU time that the job is expected to run is indicated on the CLASS card. If the CPU time exceeds this time, then the job will be pulled by the system without printing out the current temperature distribution. In order to prevent this, the maximum CPU time (seconds) for the IBM 360/91 at ORNL or IBM 360/195 at ORGDP is specified as the first entry in Card 2. As a safety factor the code subtracts five seconds from this specified time. If the code determines that the job is being run on the IBM 360/75, then it multiplies the number in this entry by 5. After each iteration or time step, a check is made to see if the CPU time exceeds the specified time. If so, it completes all of the output options which are specified and attempts to read the data for the next problem, if any. This time should be less than the CPU time specified on the CLASS card.

b. *Type Geometry*

The HEATING5 Program offers nine possible geometries (seven and eight are really the same) which are members of either the cylindrical, the Cartesian or the spherical coordinate system. These are listed below.

Cylindrical	*Rectangular*	*Spherical*
1 R-θ-Z	6 X-Y-Z	10 R
2 R-θ	7 X-Y	
3 R-Z	8 X-Z	
4 R	9 X	
5 Z		

One-, two-, or three-dimensional systems are allowed in either cylindrical or Cartesian coordinates by entering the appropriate number (1-9, as indicated above) as the second entry of Card 2. A one-dimensional model described in the radial, spherical coordinate system is defined as a Type 10 geometry.

c. *Total number of Regions*

The total number of regions of the entire configuration is entered as the third entry. A maximum of 100 regions is allowed.

d. *Total Number of Materials*

The fourth entry of this card contains the total number of different materials. There can be a maximum of 50 materials.

e. *Total Number of Materials with Change of Phase Capabilities*

The total number of materials with change of phase capabilities is the fifth entry of Card 2 (see Section 3.6.7). If phase changes are not considered in this problem, then this entry must be left blank. There can be a maximum of five such materials.

f. *Total Number of Heat Generation Functions*

The total number of different heat generation functions (a maximum of 20) is the sixth entry of Card 2. This entry must be left blank if there are no heat generation functions.

g. *Total Number of Initial Temperature Functions*

This entry (the seventh on Card 2) is the total number of different initial temperature functions, up to a maximum of 25. If there are no initial temperature functions, this entry must be left blank, and the program will assume that the initial temperature distribution is zero degrees.

h. *Total Number of Boundaries*

The eighth entry of Card 2 is the total number (maximum of 50) of boundary conditions. If there are no boundary conditions explicitly specified in the input data, this entry must be left blank.

3.6.3 *Card 3 - Input Parameters*

Each entry on this card is an integer.

a. *Gross Lattice Size*

The first three entries of this card contain the total number of gross lattice lines in (1) X or R direction, (2) Y or θ direction, and (3) Z direction. If any dimension is omitted, a zero is inserted in the appropriate entry for the total number of gross lattice lines or the appropriate entry must be left blank. There can be a maximum of 50 gross lattice lines along each axis.

b. *Total Number of Analytical Functions*

The fourth entry is the total number of different analytical functions. There can be a maximum of 25 of these functions. If there are no analytical functions, then this entry must be left blank.

c. *Total Number of Tabular Functions*

The total number of tabular functions is the fifth entry of this card. There can be a maximum of 25 functions. If there are no tabular functions, then this entry must be left blank.

d. *Temperature Units*

For problems involving radiation, the temperature units *must* be either °F or °C. Entry 6 is used only for problems involving radiation. It indicates that the units of temperature are either in degrees Fahrenheit or degrees Celsius.

Entry 6	*Temperature*
0 or Blank	°F
1	°C

e. *Three-Dimensional Output Map Flag*

The seventh entry on this card is a flag which is only used for three-dimensional problems. Normally, the temperature output map for three-dimensional problems is printed for each XY or Rθ plane. If this entry is nonzero, then the temperature output map will be printed for each XZ or RZ plane.

f. *Updating Temperature-Dependent Properties*

The eighth entry on this card specifies the number of iterations which are allowed before the temperature-dependent thermal properties are reevaluated for steady-state problems. Once the convergence criterion has been satisfied, the code continues to iterate. However, the temperature-dependent thermal properties are now reevaluated after every iteration until the convergence criterion is satisfied a second time. Some nonlinear problems will converge in fewer iterations if the thermal properties are not evaluated at each iteration, and certainly, the computing time per iteration will be less. It is recommended that this parameter be on the order of 10 or 20. If it is blank or zero, then the default value is unity for nonlinear problems.

3.6.4. *Card 4 - Input Parameters*

Each entry on this card is an integer.

a. *Transient Output*

For transient problems, the output may be specified in either of two ways:

(1) The temperature distribution may be printed out at equally spaced times. To specify this option, the first entry on Card 4 must contain the number of initial time steps between outputs, and the second entry must be blank. For example, if a value of 5 is entered, the temperature distribution will be printed at times whose spacing is equal to five times the initial time step.

(2) The temperature distribution may be printed out at unequal time increments. To choose this option, the first entry on Card 4 must be left blank and the second entry must contain the number of times the temperature distribution will be printed out. The actual output times will be entered on the O cards.

The temperature distribution is automatically printed out prior to the first time step for transient calculations and prior to the first iteration for steady-state problems. In addition, the temperature distribution may be printed out up to 100 times for each transient portion of the problem. For steady-state-only calculations, the first and second entries may be left blank.

b. *Graphical Output*

The third entry on Card 4 indicates whether or not graphical output is desired. At times it is desirable to have plots of the temperature distribution to aid in the interpretation of the results from HEATING5. By entering a nonzero integer in this field, a data set will be created which contains the temperature distributions along with certain parameters which identify the problem. The absolute value of the integer identifies the unit number on which the data set is to be created. If it is a positive integer, a new data set is created. If it is negative, then the temperatures from this run will be added to those from a previous run. The data set containing the temperatures from the previous run will be read from the unit whose number is one less than the absolute value of this entry (or one less than the unit number of the data set to be created). This data set can then be used by a plotting package to create various plots

to include temperatures versus time, temperature profiles and isothermal plots. If this option is invoked, the user must supply a DD card to define the data set. The records are variable in length. The longest and most frequently written record is 8(N + NBC) bytes long where N is the number of nodes and NBC is the number of boundary conditions. See Appendix A for a sample DD card.

c. *Frequency of Output for Plots*

This entry, the fourth on Card 4, specifies the number of time steps between each output of the temperature distribution on the data set defined in Section 3.6.4(b). If this entry is blank or zero and if the preceding entry is nonzero, then the temperature distribution will be written on the data set each time that a normal printout as defined in Section 3.6.4(a) occurs.

d. *Special Monitoring of Temperatures*

The user may wish to tabulate the temperatures of a few nodes as a function of the number of iterations or time steps. In this way one can keep track of what is happening at a few nodes of interest without getting excessive output by having to print out the entire temperature distribution. This output option is in addition to the standard output of the temperature distribution. To invoke this option, the number of iterations between printouts for steady-state calculations or the number of time steps between printouts for transient calculations is entered in the fifth field of Card 4. If this entry is nonzero, the node numbers whose temperatures are to be printed out are specified on the S cards. If this option is not desired, this entry is left blank.

e. *Initial Temperature Input Unit*

The sixth entry on this card specifies the unit number from which the explicitly specified lattice-point initial temperatures are read. If the entry is a positive integer, then it specifies the unit number from which the initial temperatures are read in formatted form. The records vary in length with the maximum size being 80 characters. If it is a negative integer, then its absolute value specifies the unit number from which the initial temperatures are read in unformatted form. If there are no initial temperatures explicitly specified, then this entry is left blank. If the unit specified is other than the standard card input, the user must insure that the appropriate DD card has been supplied to describe the data set.

f. *Final Temperature Output Unit*

In addition to the normal output, the user may wish to have the final temperature distribution saved in some manner to facilitate the restarting of the problem. If this option is desired, the unit number on which the final temperature distribution is to be written is entered on the seventh entry on Card 4. If the entry is a positive integer, then it specifies the unit number on which the final temperature distribution is to be written in formatted form. The records vary in length with the maximum size being 80 characters. If the entry is a negative integer, then its absolute value specifies the unit number on which the final temperature distribution is to be written in unformatted form. If the user does not wish to save the final temperature distribution, then this entry must be left blank. If this entry is nonzero, the user must see that the appropriate DD card has been supplied to correctly identify the specified unit (see Appendix A).

As an example, if the user feels that he may wish to restart the problem at some future date, he may enter the unit number for the standard card punch (7) as the seventh entry on Card 4. Then, when the user actually restarts the problem, he will enter the standard input unit (5) in entry 6 on Card 4. The punched output from the previous run (the final temperature distribution) will be the IT deck in the current run.

g. *Problem Status Unit for Remote Users*

This feature is designed to allow remote users to determine the status of their problem without having to wait for normal computer turnaround. If a positive number is entered in the eighth field of Card 4, then this designates the unit number on which error messages and selected information concerning the status of the problem is written. The user must supply the appropriate DD card to define the specified unit. As an example, the user may wish to use this option to have information stored on his PDP-10 file. It can then be printed out on a remote terminal. One can locate data errors and resubmit the corrected problem without having to wait for the main listing from the computer. Using this concept, one could possibly speed up turnaround time.

3.6.5. *Card 5 - Input Parameters*

Each entry on this card is a floating-point number except for entries 1, 2, and 6, which are integers.

a. *Type of Problem*

The first entry on Card 5 specifies the type of problem. It may be steady-state only, transient only, or combinations of steady-state and transient calculations. The number to be entered is in accordance with the following list:

1	S-S only	-1	trans. only
2	S-S, trans.	-2	trans., S-S
3	S-S, trans., S-S	-3	trans., S-S, trans.
4	S-S, trans., S-S, trans.	-4	trans., S-S, trans., S-S
.		.	
.		.	
.		.	
n	S-S, trans., S-S, trans. S-S, etc.	-n	trans., S-S, trans., S-S trans., etc.

If, for example, a 3 is entered, the program will first perform a steady-state calculation at time zero; next the transient calculation; then a steady-state calculation at the final transient time using the final transient temperatures as the initial guess for the steady-state temperatures. Any number of combinations is allowable; however, machine running time should be considered. (Note, careful consideration must be given to formation of time-dependent and heat generation functions with a problem type which uses more than one transient and one steady-state pass.)

b. *Maximum Number of Steady-State Iterations Allowed*

If the maximum number of steady-state iterations is reached, and the convergence criterion is not satisfied, then the program will write "END STEADY-STATE CALCULATIONS, CONVERGENCE NOT SATISFIED" on the printer and will terminate the calculation and call for the next problem. Normally 200 to 500 iterations are sufficient to converge to the solution. In the event that the convergence criterion is not satisfied, one may wish to save the final temperature distribution in order to restart the problem as discussed in Section 3.6.4(f). This entry is the second of Card 5 and is left blank for pure transient problems. If it is zero or blank for steady-state problems, then a default of 500 is used.

c. *Steady-State Convergence Criterion*

This entry, which contains the value of ϵ in Eq. (2.14), affects the steady-state type of calculation and may be left blank for a transient-only problem. The steady-state calculation will continue until the convergence criterion is met. Since the criterion which insures convergence varies from case to case, the user must rely on his own judgment and experience in determining the correct value for his particular problem. If it is left blank for steady-state problems, the code assumes $\epsilon = 10^{-5}$. This is the third entry of Card 5.

d. *Steady-State Over-Relaxation Factor*

The value of β [see Eq. (2.13)] is the fourth entry on this card and its value must be in the range $1 \leqslant \beta < 2$. This entry may be left blank for transient-only problems. If it is left blank for steady-state problems, the code assumes that the initial value of β is 1.9.

e. *Time Increment*

This entry (the fifth on Card 5) contains the initial time increment, Δt [see Eq. (2.19)], for transient problems that will be solved by one of the explicit techniques. For transient problems which will be solved using the implicit procedure, this entry must be left blank. For the Classical Explicit Procedure, the time increment must satisfy certain criteria to insure stability [see Eq. (2.20)]. The HEATING5 Program calculates the stability criterion and writes, "THE STABILITY CRITERION IS (the value of criterion) FOR POINT (lattice point yielding the above value)". If the input time increment is less than or equal to the stability criterion, then the code writes, "THE INPUT TIME INCREMENT SATISFIES THE STABILITY CRITERION," and uses the input value. If the input time increment is too large (but not more than a factor of 10 too large), the program sets Δt equal to the calculated stability criterion and writes, "THE INPUT TIME INCREMENT DOES NOT SATISFY THE STABILITY CRITERION–THE TIME INCREMENT IS NOW = (value of Δt used in subsequent calculations)." If the input Δt is too large by more than a factor of 10, the program writes, "CASE DELETED–INPUT TIME INCREMENT EXCEEDS THE STABILITY CRITERION BY MORE THAN A FACTOR OF 10," and goes on to the next problem. For steady-state-only problems, this entry may be left blank.

f. *Levy's Explicit Method Option*

The sixth entry (an integer) is the factor by which the stable time increment is multiplied to form the time increment for Levy's explicit method. If the entry is blank or less than 2, then Levy's method will not be used. If Levy's explicit method is specified, then the problem is not terminated if the input time increment is more than an order of magnitude larger than the stability criterion as noted in Section 3.6.5(e). Instead, the time increment is initially set equal to the stability criterion. HEATING5 always executes 10 time increments using a stable time step prior to using Levy's explicit method. This entry may be left blank for steady-state-only (Type 1) problems.

g. *Initial Time*

This entry affects both transient and steady-state problems. It indicates the initial time for problems with a negative type and the time at which the time-dependent functions are evaluated for problems whose types are greater than zero. For example, assume that a type 4 problem is run. Let t_o represent the initial time (entry 7) and t_f represent the final time (entry 8). Then, the time-dependent functions will be evaluated at t_o for the first steady-state calculation. The first transient calculation will begin at time t_o, using the results of the steady-state calculations as the initial temperatures, and will stop at t_f. The second steady-state calculation will then be performed, and the time-dependent functions will be evaluated at t_f. The results from the transient calculation will be used as the first approximation to the steady-state temperature distribution. Finally, the last transient calculation will begin at t_f and continue until $t_f + (t_f - t_o)$. The value $(t_f - t_o)$ will be added to the output times from the first transient calculation to obtain the output times for the second transient case. If entry 7 is left blank, then t_o will be zero.

h. *Final Time for Transient Calculation*

The final time for the first transient calculation is specified as the eighth entry on Card 5. For steady-state problems, this entry may be left blank.

3.6.6. *Region Data (Cards R1 and R2)*

Each region is described by two cards which must appear in pairs. The cards are repeated for each region. The number of pairs of cards is the third entry on Card 2. There must be at least one region for each problem.

a. *Card R1*

(1) Region Number

This entry contains the number of the region to be described. Regions are to be numbered consecutively beginning with number 1 up to a maximum of 100 regions. The region numbering system does *not* require that a region occupy any particular zone in the overall configuration. In the example of Fig. 3.1, the left-most region could be numbered Region 2, and the right-most region, Region 1. Refer to Section 3.2 for further details about regions. This entry is the first on Card R1 and must be an integer.

(2) Material in Region

The second entry of this card indicates by an integer the number of the material which occupies the region named in the first entry of this card. This entry is left blank if the region does not contain a material (gap region).

(3) Region Dimensions

Dimensions of the region boundaries are entered as floating-point numbers and are arranged in the following order:

a) smaller dimension of X or R region boundary
b) larger dimension of X or R region boundary
c) smaller dimension of Y or Θ region boundary
d) larger dimension of Y or Θ region boundary
e) smaller dimension of Z region boundary
f) larger dimension of Z region boundary

These dimensions are entered as the third through eighth entries, respectively. If the problem is one- or two-dimensional, then the region dimensions for the corresponding unnecessary coordinate or coordinates are omitted. The region dimensions must be nonnegative. Although it is usually more convenient to place the overall configuration at the origin in X-Y-Z geometry, it is not necessary to do so; however, all dimensions are entered as their distance from the origin. In R-θ-Z geometry, R = 0.0 must be located at the origin with R increasing radially; Z is not necessarily begun at the origin, but must extend upward; and θ lines or planes increase clockwise in order to be compatible with the X-Y-Z system.

b. *R2 Card*

This card must be included in conjunction with the appropriate R1 card, even if it is blank.

(1) Initial Temperature of Region

The initial temperature function number of the region specified by the first entry of Card R1 is entered as an integer. If this entry is left blank, then the program assumes that the initial temperature for the region is zero. This entry is left blank for a gap region.

(2) Heat Generation of Region

This entry contains the number (an integer) of the heat generation function associated with the region given on the first card of this pair. If this entry is left blank, then the code assumes that the region does not generate heat. This entry is left blank for a gap region.

(3) Boundary Numbers

The remaining entries of this card are the boundary numbers defining the boundary conditions corresponding to the six boundaries of the region described by the first card of this pair. These numbers are integers and are entered as follows:

Each entry contains the boundary number of the region boundary appearing in the corresponding entry of Card R1. A boundary condition cannot be specified on a boundary dividing two regions unless it is a type 3 boundary condition or unless one of the regions is a gap region. For one- or two-dimensional cases, region dimensions are not specified for the unnecessary coordinate or coordinates, and the corresponding boundary numbers are left blank. The entry is also left blank for boundaries which are insulated.

3.6.7. *Material Data (Cards M and PC)*

A group of cards consisting of an M card and possibly a PC card is required to describe each material. The total number of groups is the fourth entry on Card 2. There can be five materials with change of phase capabilities, and they must be the first ones described on the M cards. If the thermal conductivity of a material is anisotropic, then it is specified as being temperature-dependent, and the associated temperature-dependent function is specified as being user-supplied. The user then programs the anisotropic algorithm for that material in subroutine CONDTN. The following labeled common must be added to subroutine CONDTN:

```
COMMON /THBSBC/ NBDTP, NDIR
```

The variable NDIR will contain a value of 1, 2, or 3 indicating the thermal conductivity is to be evaluated along the X (or R), Y (or θ) or Z axis, respectively. Both variables NBDTP and NDIR are INTEGER*4.

a. *Card M*

(1) Material Number

The first entry, an integer, contains the number of the material which is to be described. Materials are numbered consecutively (each different material has a number) beginning with number 1 up to a maximum of 50 materials.

(2) Material Name

The second entry, which must begin in Column 11 and may extend through Column 18, contains the name of the material. This name, which may consist of up to eight alphanumeric characters, is used to aid in identification of output data.

(3) Constant Thermal Properties

Entries 3, 4, and 5 are floating-point numbers and respectively contain the constant thermal conductivity, constant density, and constant specific heat of the material. These entries correspond to the factor, P_o, in Eq. (3.4). Since the density and specific heat are not used in steady-state calculations, entries 4 and 5 may be left blank for type 1 problems.

(4) Temperature-Dependent Thermal Properties

Entries 6, 7, and 8 (integers) identify the analytical or tabular functions describing the thermal conductivity, density, and specific heat, respectively, as a function of temperature. These entries correspond to subscript l in Eq. (3.5). Entries 7 and 8 may be left blank for type 1 problems.

b. *Card PC*

For materials which can undergo a phase change, the phase-change or transition temperature and the corresponding latent heat are entered as floating point numbers in the first and second fields, respectively, of Card PC. The following conventions must be adhered to in describing materials with change of phase capabilities: these materials must be the first ones described on the M cards; there can be a maximum of five such materials; and the PC card is omitted for those materials which do not undergo a change of phase. The total number of materials which involve a change of phase is the fifth entry on Card 2.

3.6.8. *Heat Generation Function Data (Card G)*

Each different heat generation function is numbered, beginning with number 1, consecutively up to a maximum of 20 such functions. As indicated in Table 3.1, the heat generation function associated with a region may be dependent on position, time and temperature. Thus, the heat generation function data cards indicate the function number, the volumetric heat generation rate [corresponds to P_o in Eq. (3.4)], the time-dependent function parameter, the temperature-dependent function parameter, the X- or R-dependent function parameter, the Y- or θ-dependent function parameter, and the Z-dependent function parameter, arranged in that order, respectively. The parameters in entries 3 through 7 refer to analytical or tabular functions defining *l*, m, i, j, and k, respectively, in Eq. (3.5). If the heat generation rate per unit volume does not vary along an axis or if the problem is one- or two-dimensional, then the position-dependent function parameter corresponding to that coordinate will be left blank, and the associated function value will be set equal to 1.0. If the heat generation rate per unit volume does not vary with time or temperature, then the time- or temperature-dependent function parameter will be omitted and the associated function value or values will be set to 1.0. The first entry on this card is an integer, the second is a floating point number and the remaining five are integers. The total number of cards is indicated by the sixth entry on Card 2. The heat generation rate for a region may be positive (heat source) or negative (heat sink). The G cards are omitted if entry 6 on Card 2 is blank or zero.

3.6.9. *Initial Temperature Function Data (Card I)*

Each different initial temperature function is given a number. Beginning with number 1, the initial temperature functions are numbered consecutively up to a maximum of 25. Since the initial temperature function associated with a region can be a function of position, then this data card consists of five entries. All of them are integers except the second which is a floating point number. The first entry contains the initial temperature function number. The second entry contains the constant factor describing the initial temperature function. This term corresponds to P_0 in Eq. (3.4). The remaining entries, which identify analytical or tabular functions, contain the X- or R-, Y- or θ-, and Z-dependent function parameters corresponding to the subscripts i, j, and k, respectively, in Eq. (3.5). If the problem is one- or two-dimensional or if the initial temperature does not vary along a particular axis of the region, then the position-dependent function parameter associated with the coordinate will be left blank, and the corresponding function value will be set equal to 1.0. The total number of cards is the seventh entry on Card 2, and if this entry is blank or zero, then the I cards are omitted, and the initial temperature distribution is assumed to be zero.

3.6.10. *Boundary Data (Cards B1, B2, B3, and B4)*

Excluding insulated or contact type boundaries, each unique boundary is numbered consecutively up to a maximum of 50. The B1 and B2 cards are omitted if the eighth entry on Card 2 (this entry indicates the total number of boundary conditions specified) is blank or zero.

a. *Card B1*

(1) The first entry on Card B1 is an integer and contains the boundary number.

(2) The second entry (an integer) indicates the type of boundary. HEATING5 offers three boundary types which are numbered 1, 2, or 3, corresponding to the following:

1 implies surface-to-boundary,
2 implies prescribed surface temperature, and
3 implies surface-to-surface.

If this entry is blank or zero, then no heat transfer connections will be made and the boundary will be treated as an insulated boundary.

(3) The third entry, a floating-point number, contains the boundary temperature, T_b [corresponds to P_0 in Eq. (3.4)]. This entry is left blank for a Type 3 boundary condition.

(4) Since the boundary temperature can be a function of time, the fourth entry on this card contains the time-dependent parameter (an integer) which corresponds to subscript l in Eq. (3.5). This parameter identifies an analytical or tabular function. If the boundary temperature is independent of time or if the boundary type is 3, then this entry will be left blank.

b. *Card B2*

Each entry on Card B2 is a floating-point number except Entry 6 which is an integer. This card is left blank for a Type 2 boundary condition.

(1) Entry 1 contains the heat transfer coefficient, h_c, for forced convection. [Corresponds to P_O in Eq. (3.4).]

(2) Entry 2 contains the coefficient h_r, for radiation. [Corresponds to P_O in Eq. (3.4).]

(3) Entry 3 contains the coefficient, h_n, for natural convection [Corresponds to P_O in Eq. (3.4).]

(4) The fourth entry contains the exponent, h_e, for natural convection (or other nonlinear heat transfer process). [Corresponds to P_O in Eq. (3.4).]

(5) The fifth entry contains the prescribed heat flux, h_f, across the boundary. [Corresponds to P_O in Eq. (3.4).]

(6) The time- and temperature-dependent parameter flag, an integer, is the sixth and final entry on the B2 card. If any of the five preceding parameters are functions of time or temperature, then additional information must be entered on B3 and/or B4 cards. The time- and temperature-dependent flag indicates whether or not the B3 and B4 cards are present for this particular boundary condition. Its value is determined according to the following table:

Entry Six	*Additional Cards*
0	None
1	B3 Only
2	B4 Only
3	B3 and B4

c. *Card B3*

All five entries on Card B3 are integers. Each integer identifies the analytical or tabular function that defines the time-dependent function associated with the respective parameter on Card B2. Each entry corresponds to subscript l in Eq. (3.5). If an entry is zero, then the associated parameter is not time-dependent.

d. *Card B4*

This card is just like Card B3 except each integer identifies the analytical or tabular function that defines the temperature-dependent function associated with the respective parameter on Card B2. Each entry corresponds to subscript m in Eq. (3.5).

In evaluating a particular parameter, the code uses Eq. (3.4) where P_O is the appropriate value from Card B2, $F_l(t)$ is the time-dependent function as defined by Card B3 and $F_m(T)$ is the temperature-dependent function as defined by Card B4.

The B1 and B2 cards must appear in pairs, and a pair is entered for each boundary. B3 and B4 cards, if any, must follow their respective B1 and B2 cards.

3.6.11 *Lattice Description (Cards L1, N1, L2, N2, L3 and N3)*

For each axis, gross lattice data are entered on two sets of cards, the first set specifying the lattice dimensions and the second indicating the mesh division between gross lattice lines. All of the numbers on the cards on the first set (L cards) are of the floating-point type and are entered, by specifying all of the gross lattice dimensions in each direction, sequentially on one or more cards. The cards of the second set (N cards) specify the number (an integer $\geqslant 1$) of equal increments which are between the gross lattice lines whose dimensions are given on the cards of the first set. In particular, an entry on an N card specifies the number of equal increments between the gross lattice line in the corresponding entry on the related L card and the gross lattice line immediately following it. This procedure is repeated for each coordinate. For degenerate geometries, the corresponding unnecessary sets of cards must be omitted.

a. *Card L1*

The L1 cards correspond to the X or R coordinate, and the number of entries corresponds to Entry 1 of Card 3. If there are more than 8 entries, subsequent cards are used.

b. *Card N1*

The N1 cards correspond to the X or R coordinate. There will be one less entry here than there are on the L1 cards. Additional cards are used for more than eight entries.

c. *Card L2*

The L2 cards correspond to the Y or θ coordinate, and the number of entries corresponds to Entry 2 of Card 3. If there are more than eight entries, subsequent cards are used.

d. *Card N2*

The N2 cards correspond to the Y or θ coordinate. There will be one less entry here than there are on the L2 cards. Additional cards are used for more than eight entries.

e. *Card L3*

The L3 cards correspond to the Z. coordinate, and the number of entries corresponds to Entry 3 of Card 3. If there are more than eight entries, subsequent cards are used.

f. *Card N3*

The N3 cards correspond to the Z coordinate. There will be one less entry here than there are on the L3 cards. Additional cards are used for more than eight entries.

3.6.12. *Analytical Function Data (Cards A1 and A2)*

Each analytical function, defined in Eq. (3.1), is described by an A1 card and one or more A2 cards.

a. *Card A1*

Each different analytical function is numbered, and there can be a maximum of 25 such functions. The first entry on Card A1 is the unique analytical function number, an integer. The second entry, also an integer, is the number of coefficients, A_i, which are on the A2 cards. If this entry is blank or zero, then the code assumes that a user-supplied function will be supplied for the parameter which uses this particular analytical function.

b. *Card A2*

The A2 cards contain from one to four ordered pairs, where each ordered pair is defined as follows: the first element of an ordered pair consists of an integer i; the second element consists of the value of the coefficient A_i. The A2 cards will be continued until each coefficient in the analytical function is defined. If the second entry on Card A1 is blank or zero, then the related A2 card is omitted.

As an example, the cards which are necessary to describe $f(x) = 100\sin 5x + \exp(-3x)$ are presented in Table 3.2.

Table 3.2. A Cards Necessary to Describe the Function $100\sin 5x + \exp(-3x)$.

Column	9	18	27	36	45	54	63	72	75
	1	4							A1
	8	100.0	9	5.0	6	1.0	7	-3.0	A2

The total number of analytical functions is the fourth entry on Card 3. The A cards are omitted if this entry is blank or zero.

3.6.13. *Tabular Function Data (Cards T1 and T2)*

Each tabular function is numbered consecutively beginning with one up to a maximum of 25 functions. The tabular function is assumed to be a set of linearly connected points. The function is described by specifying a set of ordered pairs. Each ordered pair contains an independent variable and its functional value; viz., t_1, $G_f(t_1)$; t_2, $G_f(t_2)$; etc. A maximum of 25 points (pairs) is allowed, and linear interpolation is performed between the points by the program. The values of t_i must be entered in ascending order.

a. *Card T1*

The first entry on the T1 card (an integer) is the tabular function number. The number of points (an integer) is the second entry.

b. *Card T2*

The T2 card contains the first four ordered pairs, all floating point numbers. If there are more than four ordered pairs in the function, they are entered on subsequent T2 cards.

Figure 3.6 is an example of an acceptable tabular function. The input for this example is presented in Table 3.3.

Fig. 3.6. Plot of a Sample Tabular Function

Table 3.3. T Cards Necessary to Describe the Sample Tabular Function Depicted in Fig. 3.6.

Column	9	18	27	36	45	54	63	72	Card
	1	9							T1
	0.0	25.0	1.0	30.0	2.0	30.0	2.5	35.0	T2
	3.5	35.0	4.0	30.0	4.5	30.0	5.5	20.0	T3
	5.501	10.0							T4

The total number of tabular functions is the fifth entry of Card 3. The T cards are omitted if this entry is blank or zero.

3.6.14. *Output Times (Card O)*

Each entry on this card is a floating-point number. Since the second entry on Card 4 indicates the total number of output times which are to be read, the O cards are omitted if this entry is blank or zero. The transient output times are entered in chronological order on the O cards. There can be eight entries per card, and the O card is repeated as often as necessary to describe the output array. The maximum number of output times is 100, not counting the automatic printout which occurs prior to the initial time step.

3.6.15. *Node Numbers for Special Monitoring of Temperatures (Card S)*

As optional output, one may specify up to 20 nodes whose temperatures will be printed out as a function of the number of iterations for steady-state calculations or the number of time steps for transient calculations. The first entry on Card S contains the total number of nodes whose temperatures are to be tabulated. The remaining fields contain the actual node numbers. If more than seven nodes are specified, their numbers will appear on additional cards. All entries on the S cards are integers. The frequency for printing out the temperatures of such nodes appears as the fifth entry on Card 4, and the S card or cards are omitted if this entry is blank or zero.

3.6.16. *Initial Temperatures and Melting Ratios*

These cards (or card images) are generated as output by HEATING5 if a positive number appears in Entry 7 on Card 4; and, generally, they are used only when restarting a job by merely inserting the generated deck at this location in the original deck and resubmitting the job. As noted in Section 3.6.4(e), one must enter the unit number of the card reader in the sixth field of Card 4. Entry 6 on Card 4 specifies the unit on which these data are to be read. If the unit specified is other than the standard card input, the user must insure that the appropriate DD card has been supplied to describe the unit. If Entry 7 on Card 4 is nonzero,

then the code generates these data at the end of a problem and writes it on the unit specified in Entry 7. However, the user must insure that the appropriate DD card has been supplied to correctly identify the unit specified on Entry 7. If Entry 6 on Card 4 is blank or zero, then these cards are omitted.

Since the user may wish to explicitly specify the initial temperature or melting ratio at some point or points, a description of these data is given below.

a. *Job Description (Card IT1)*

This card image gives a descriptive title and can contain alphanumeric characters in the first 72 columns. The card may be blank but it cannot be omitted.

b. *Initial Time and Lattice Point Numbers (Card IT2)*

The first entry, a floating point number which occupies the first ten columns on the IT2 card, specifies the initial problem time. This value overrides the initial time which appears on Entry 7 of Card 5. Normally, this deck will have been generated by the code on a previous run. Thus, the code supplies this value as the time for which the following temperature distribution and melting ratios occur. If the user generates the IT deck, then he must insure that the initial problem time is entered here. The second entry on the IT2 card is an integer and contains the total number of lattice points whose initial temperatures are explicitly specified. It occupies columns 11 through 15 of Card IT2. The third entry, also an integer, contains the total number of nodes whose initial melting ratios are explicitly specified. It also occupies a five-column field, namely columns 16 through 20.

c. *Lattice Point Temperatures (Card IT3)*

The IT3 card can contain up to five pairs of numbers with each pair defined as follows:

(1) The first member of the pair is a lattice point number. It is an integer and occupies a five-column field.

(2) The second member is the initial temperature of the lattice point whose number appears in the first member. It is a floating-point number and occupies a ten-column field.

The number of pairs to be entered on the IT2 cards is specified on entry 2 of Card IT2. Card IT3 is repeated until all pairs have been described. The temperatures specified by this input data override the corresponding temperatures generated by the I cards.

d. *Lattice Point Initial Melting Ratios (Card IT4)*

The IT4 card contains initial melting ratios for each node which is currently undergoing a phase change. The format of the IT4 card is as follows:

(1) The first entry is the number of a node which is currently undergoing a phase change. It is an integer and the field occupies the first five columns.

(2) The second entry is the material number currently undergoing a phase change for the node which was defined in the previous field. This entry is an integer which occupies the sixth through the tenth columns.

(3) The third entry, a floating point number occupying the eleventh through the twentieth columns, contains the initial melting ratio for the portion of the material associated with the node defined on the first two entries of this card as currently changing phase.

The total number of lattice points with initially specified melting ratios or the total number of IT4 cards is the third entry on Card IT2.

3.6.17. *Implicit Transient Technique Parameters*

If this problem involves transient calculations (Entry 1 on Card 5 is not equal to 1) and if the implicit technique is to be used to calculate estimates to the transient temperature distribution, then the fifth entry (time step) of Card 5 must be left blank and additional data must be supplied on the IP and TP cards. If it is anticipated that similar problems will be run a large number of times such as ones arising in a parametric study, then it is recommended that the variables on the IP and TP cards be optimized since it could significantly reduce the overall computer time.

a. *Card IP*

This entire card or any of its entries may be left blank, and the default values will be used. They are based on experiences with a few two-dimensional RZ models. They are certainly not the best values that can be used for a given problem, but they are probably good starting points. There are eight entries on this card. The first five are floating point numbers, and the last three are integers.

(1) The first entry contains one of two convergence criteria which must be met in order for the iterative technique to terminate successfully at each time step. This convergence criterion corresponds to ϵ_1 in Eq. (2.37). The default is 10^{-5}.

(2) The second entry contains the second convergence criterion which corresponds to ϵ_2 in Eq. (2.38). The default for this parameter is 10^{-3} meaning the maximum normalized residual must decrease by three orders of magnitude. It has been found that for problems with an abnormally small time step (one approximately equal to the stability criterion or less) this criterion with $\epsilon_2 = 10^{-3}$ may not be satisfied. Increase the time step size, if possible. Otherwise, experiment with the value of ϵ_2.

(3) The third entry contains the convergence criterion for problems involving temperature-dependent parameters. This convergence criterion which corresponds to ϵ_5 in Eq. (2.41) is used in addition to the first two. The default is 10^{-5}.

(4) This entry, the fourth, defines the implicit technique which will be used to solve the transient problem. It refers to Θ in Eq. (2.34) and must be chosen so that $0.5 \leqslant \Theta \leqslant 1.0$. The default is 0.5.

(5) The fifth entry defines the initial value of the point successive overrelaxation iteration acceleration parameter [ω in Eq. (2.34)]. It also defines the method that will be used to update the acceleration parameter. If this entry is positive, then the acceleration parameter will remain constant throughout the calculations and will be equal to the value of this entry. If it is blank or zero, then the acceleration parameter will be optimized empirically as a function of time. This appears to be the best option for nonlinear problems. If it is negative, then the acceleration parameter will be calculated using Carré's technique (Ref. 8). The absolute value of this entry must be less than 2.0.

(6) This entry, an integer, defines the number of time steps between attempts to optimize the acceleration parameter empirically [referred to as N_ω in Section 2.3.3(d)]. It is used only when entry 5 is zero or blank. The default value is 1.

(7) For the case when the acceleration parameter will be updated empirically (Entry 5 is blank or zero), then this entry, the seventh, defines the change-in-number-of-iterations criterion [referred to as I_ω in Section 2.3.3(d)] which must be met before the acceleration parameter will be updated. The default is 5. For the case when the SOR acceleration parameter will be updated using Carré's technique, then this entry defines the number of iterations between updates. The default is 12.

(8) The last entry is the change-in-number-of-iterations criterion [referred to as J_ω in Section 2.3.3(d)] which is used to determine when a good estimate to the optimum acceleration parameter has been found. This entry is used only when the acceleration parameter will be updated empirically (Entry 5 on this card is blank or zero). The default is 2.

b. *Card TP*

When an implicit scheme is used to solve a transient problem, the time step may be variable. This allows the time step to increase as the solution smooths out and to decrease when some parameter varies rapidly with time. The information controlling the value of the time step is contained on one or more of the TP cards. The size of the time step is automatically adjusted in order to get printouts of the temperature distribution at the specified time. If the size of the coefficients in the system of equations varies by orders of magnitude (10^5 or greater), it has been observed that point-successive overrelaxation iteration may converge very slowly (it may appear to not converge at all). This occurs when the grid spacing or thermal properties vary by orders of magnitude over the problem. It can be observed by examining the stability criterion table in the output. If this appears to be happening, either further subdivide some of the larger nodes or combine some of the smaller ones. In some cases, it may help to use a larger time step size. All seven entries are floating point numbers.

(1) The first entry is the initial time step.

(2) After the temperature distribution has been calculated, the current time step is multiplied by a factor. The value of this factor is entered in the second field of the TP card. The default value is 1.0. For many problems whose parameters vary mildly with time and/or temperature, values between 1.0 and 1.3 have been acceptable.

(3) The maximum value of the time step is the third entry. Once the time step reaches this value, it is no longer increased. The default is 10^{50}.

(4) The fourth entry contains the maximum time that the time step information on this card applies. If the time reaches this value, then a new TP card is read. The default is 10^{50}

(5) The fifth entry contains the maximum temperature change allowed at a node over a time step. The time step size is adjusted according to the procedure outlined in Section 2.3.4. If this entry is blank or zero, then this feature is not invoked in calculating the time step size.

(6) The sixth entry contains the maximum percent of relative temperature change allowed at a node from one time level to the next one. The time step size is adjusted according to the procedure outlined in Section 2.3.4. If this entry is blank or zero, then this feature is not invoked in calculating the time step size.

(7) The seventh and final entry on this card contains the minimum value of the time step. Once the time step size reaches this value, it is no longer decreased. The default is one-tenth of the initial time step size.

3.6.18. *Blank Card*

If the user wishes to solve several problems with one run, he merely inserts a blank card between each problem deck.

3.7. Problem Size Limits

The problem size limits are summarized in Table 3-4.

Table 3.4. Problem Size Limits

Item	Maximum Number
Analytical Functions	25
Boundary Conditions	50
Fine Lattice Lines Along Any Axis	100
Gross Lattice Lines Along Any Axis	50
Heat Generation Functions	20
Initial Temperature Functions	25
Lattice Points	1400
Materials	50
Materials with Change-of-Phase Capabilities	5
Number of Nodes in Special Temperature Table	20
Points Per Tabular Function	25
Printout Times	100
Regions	100
Surface-to-Surface Connectors	2000
Tabular Functions	25

3.8. Summary and Format of Input

The following table is a concentrated summary of the format and information needed to prepare the input data deck. Except for the IT cards, columns 73 through 80 of each card are reserved for identification, and the user may or may not choose to punch this information. The table in this column gives proposed identification names for the cards and, in parentheses, the reference sections in the report where more information can be found. The first line in each format box includes the variable name actually used in the program and, in parentheses, the format of that variable. The rest of the box includes a short explanation of the input in that box. In some cards, where free space is available (e.g., G or I cards), additional notes have been included to describe the way the input information will be used in the program itself.

Table 3.5 Summary and Format of Input Data for HEATING5

Columns 1-9	Columns 10-18	Columns 19-27	Columns 28-36	Columns 37-45	Columns 46-54	Columns 55-63	Columns 64-72	Columns 73-80	
JOBDES(I), (18A4) Job Description								Card 1 (3.6.1)	
MXCPU (I9) Maximum CPU time for IBM 360/91 or IBM 360/195 in seconds.	NGEOM (I9) Geometry type. 1-RθZ 6-XYZ 2-Rθ 7-XY 3-RZ 8-XZ 4-R 9-X 5-Z 10-R(sphere)	NOREGT (I9) Total number of regions. (Cards R1 & R2) Maximum 100	MATLT (I9) Total number of materials. (Cards M) Maximum 50	MATSL (I9) Total number of materials with change of phase capabilities. (Cards PC) Maximum 5	NGENFN (I9) Total number of heat generation functions. (Cards G) Maximum 20	NINTFN (I9) Total number of initial temperature functions. (Cards I) Maximum 25	NBDTPT (I9) Total number of boundary conditions. (Cards B1 & B2) Maximum 50	Card 2 (3.6.2)	
IGT (I9) Total number of gross lattice lines in X or R direction. (Entries in L1) Maximum 50	JGT (I9) Total number of gross lattice lines in Y or θ direction. (Entires in L2) Maximum 50	KGT (I9) Total number of gross lattice lines in Z direction. (Entries in L3) Maximum 50	NOANAT (I9) Total number of analytical functions. (Cards A) Maximum 25	NOTBLT (I9) Total number of tabular functions. (Cards T) Maximum 25	IDEGRE (I9) Temperature units. If °C, and radiation is involved, enter 1; otherwise, leave blank.	IOPLNE (I9) Three-dimensional output map flag. Blank implies XY planes, non zero implies XZ planes.	MCOUNT(I9) Number of iterations between evaluation of temperature-dependent thermal properties for steady state cases. (Default 1)	Card 3 (3.6.3)	
NDTA (I9) Number of time steps between printed outputs. Next entry must be blank.	NPRINT (I9) Number of times output will be printed. (Entries in ∅ card), preceding entry must be blank. Maximum 100	ITAPE (I9) |ITAPE| is unit no. for plots. ITAPE=0 implies no plots. ITAPE < 0 implies plot data will be added to data from previous run read from unit no. |ITAPE| - 1.	NOUTPT (I9) Number of time steps between outputs for plots on unit number |ITAPE|. If blank, output is at normal printouts.	NTS (I9) Number of iterations or time steps between output for special monitoring of temperatures. (Card S)	JIN (I9) Unit number for reading initial temperature distribution. Positive, formatted. Negative, unformatted. (5 on IBM for cards)	JOUT (I9) Unit number for writing out final temperature distribution. Positive, formatted. Negative, unformatted. (7 on IBM for cards)	IERROR (I9) Problem status unit for remote users.	Card 4 (3.6.4)	
NTYPE (I9) Problem type 1 steady state; -1 transient. (See Section 3.6.5(a) for other types)	NOITX (I9) Maximum number of steady state iterations. (Default, 500)	EPI (E9.0) Steady state convergence criterion. (Default, 10^{-5})	BETA (E9.0) Steady state over-relaxation factor. $1.0 \leq \beta < 2.0$ (Default, 1.9)	DELTAT (E9.0) Time increment for transient problem involving explicit technique. Must be left blank for implicit technique.	KTMFCT (I9) Factor by which stable time increment is increased if Levy's explicit method is used.	TIM (E9.0) Initial time. (See Section 3.6.5(g) for more information)	FTIME (E9.0) Final time.	Card 5 (3.6.5)	
NOREG (I9) Region number. Maximum 100	MATL (I9) Region material number. (Card M) Leave blank for gap region.	RRIN (E9.0) Smaller X or R region dimension.	RROT (E9.0) Larger X or R region dimension.	THLT (E9.0) Smaller Y or θ region dimension.	THRT (E9.0) Larger Y or θ region dimension.	ZZBK (F9.0) Smaller Z region dimension.	ZZFR (E9.0) Larger Z region dimension.	R1	Coupled (Sec. 3.6.6)
ITEM (I9) Region initial temperature function number. (Card I)	NGEN (I9) Region heat generation function number. (Card G)	NBDIN (I9) Boundary condition on smaller X or R. (Card B)	NBDOT (I9) Boundary condition on larger X or R. (Card B)	NBDLT (I9) Boundary condition on smaller Y or θ. (Card B)	NBDRT (I9) Boundary condition on larger Y or θ. (Card B)	NBDBK Boundary condition on smaller Z. (Card B)	NBDFR (I9) Boundary condition on larger Z. (Card B)	R2	
MAT (I9) Material number. Maximum 50	Column 10 Blank: MATNAM (A8) Material name.	CONDUC (D9.0) Conductivity if constant.	DENSTY (D9.0) Density if constant. Leave blank for steady state only.	SPHEAT (D9.0) Specific heat if constant. Leave blank for steady state only.	NCONTP (I9) Conductivity temperature-dependent function number.	NDENTP (I9) Density temperature-dependent function number. Leave blank for steady state only.	NSPHTP (I9) Specific heat temperature-dependent function number. Leave blank for steady state only.	M	
					NOTE: For the 6th through 8th entries, positive integer implies analytical function and negative integer implies tabular function.				
SLIM (D9.0) Phase-change or transition temperature.	SLHM (D9.0) Latent heat.	NOTE: This card is only present for materials with change of phase capabilities. There can only be five such materials and they must be the first ones described on the M cards.						PC	
NGN (I9) Heat generation function number. Maximum 20	GENO (D9.0) Volumetric heat generation rate, if constant.	NGNTM (I9) Time-dependent function number.	NGNTP (I9) Temperature-dependent function number.	NGNR (I9) X- or R-dependent function number.	NGNTH (I9) Y- or θ-dependent function number.	NGNZ (I9) Z-dependent function number.	NOTE: For the 3rd through 7th entries, positive integer implies analytical function and negative integer implies tabular function.	G (3.6.8)	
INTM (I9) Initial temperature function number. Maximum 25	TEMPIN (D9.0) Initial temperature, if constant.	ITMR (I9) X- or R-dependent function number.	ITMTH (I9) Y- or θ-dependent function number.	ITMZ (I9) Z-dependent function number.	NOTE: For the 3rd, 4th and 5th entries, positive integer implies analytical function, negative integer implies tabular function.			I (3.6.9)	
NBDTP (I9) Boundary number. Maximum 50	NBYTYP (I9) Boundary type. 1-surface-to-boundary 2-isothermal 3-surface-to-surface	BYTEMP (E9.0) Boundary temperature. T_0 (Blank for type 3)	NBYTFN (I9) Time dependent function number (ℓ) for boundary temperature. (Blank for type 3)	Boundary temperature will be established as $T_b(t)=T_0 \cdot T_\ell(t)$ where $T_\ell(t)$ is the ℓth analytical function for $t > 0$, and the $(-\ell)$th tabular function for $t < 0$. NOTE: The heat flux will be established as: $q'' = h_f + h_c\Delta T + h_r(T_1^4 - T_2^4) + h_n\lvert\Delta T\rvert^{h_e}(\Delta T)$ where $\Delta T = T_1 - T_2$				B1	Coupled (3.6.10)
BHCONV (E9.0) Forced convective heat transfer coefficient. h_c	BHRAD (E9.0) Radiative coefficient. $h_r = F \cdot \sigma$ σ= Stefan-Boltzman constant F= Radiation shape factor	BHNAT (E9.0) Natural convective coefficient. h_n	BHEXP (E9.0) Natural convective exponent. h_e	BFLUX (E9.0) Prescribed heat flux (positive if going to surface) h_f	IBHFLG (I9) Parameter flag. 0-no additional cards 1-B3 card only 2-B4 card only 3-B3 & B4 cards			B2	
NBHCTM (I9) Forced convective heat transfer coefficient time-dependent function.	NBHRTM (I9) Radiative coefficient time-dependent function.	NBHNTM (I9) Natural convective coefficient time-dependent function.	NBHETM (I9) Natural convective exponent time-dependent function.	NBHFTM (I9) Prescribed heat flux time-dependent function.	Positive integer implies analytical function. Negative integer implies tabular function.			B3	
NBHCTP (I9) Forced convective heat transfer coefficient temperature-dependent function.	NBHRTP (I9) Radiative coefficient temperature-dependent function.	NBHNTP (I9) Natural convective coefficient temperature-dependent function.	NBHETP (I9) Natural convective exponent temperature-dependent function.	NBHFTP (I9) Prescribed heat flux temperature-dependent function.	Positive integer implies analytical function. Negative integer implies tabular function.			B4	
RG (E9.0) Smallest X or R gross lattice line dimension.	RG (E9.0) Next X or R gross lattice line dimension.	RG (E9.0) Next X or R gross lattice line dimension.	Can be repeated for up to 50 gross lattice lines (but equal to entry one in Card 3).					L1 (3.6.11)	
NDRG (I9) Number of DIVISIONS between corresponding X or R gross line and the following line.	NDRG (I9)	NDRG (I9)	Must have one less entry than in L1 cards (maximum of 100 fine lattice lines in each direction).					N1 (3.6.11)	
THG (E9.0) Y or θ	Same as L1 cards except for Y or θ direction.							L2 (3.6.11)	
NDTHG (I9) Y or θ	Same as N1 cards except for Y or θ direction.							N2 (3.6.11)	
ZG (E9.0) Z	Same as L1 cards except for Z direction							L3 (3.6.11)	
NDZG (I9) Z	Same as N1 cards except for Z direction.							N3 (3.6.11)	
NANALT (I9) Analytical function number. Maximum 10	NPARM (I9) Number of coefficients in A2 cards. Maximum 9	NOTE: A1 and A2 establish the mth analytical function in the form $F_m(V) = A_{m,1} + A_{m,2}V + A_{m,3}V^2 + A_{m,4}\cos(A_{m,5}V) + A_{m,6}\exp(A_{m,7}V) + A_{m,8}\sin(A_{m,9}V) + A_{m,10}\ln(A_{m,11}V)$.							A1	Coupled (3.6.12)
NPFM (I9) Coefficient index, i.	A(NPFM) (E9.0) Coefficient value, $A_{m,i}$.	NPFM (I9) Coefficient index, i.	A(NPFM) (E9.0) Coefficient value, $A_{m,i}$.	NOTE: Can be repeated for up to 11 coefficients. NOTE: To indicate a user-supplied function, the second entry on Card A1 is left blank and Card A2 is omitted.				A2	
NTABL (I9) Tabular function number. Maximum 20	NTBPRS (I9) Number of pairs in table. Maximum 25	NOTE: The code uses linear interpolation to obtain the value of this function.						T1	Coupled (3.6.13)
ARG(1) (E9.0) Independent variable.	VAL(1) (E9.0) Corresponding function value.	ARG(2) (E9.0) Independent variable.	VAL(2) (E9.0) Corresponding function value.	NOTE: Can be repeated for up to 25 pairs.				T2	
PRTIME (E9.0) First time printout is desired.	PRTIME (E9.0) Second time printout is desired.	PRTIME (E9.0) Third time printout is desired.	Can be repeated for up to 100 printout times (but equal to entry two in Card 4).					O (3.6.14)	
NSN (I9) Total number of nodes for special monitoring of temperatures. Maximum 20	NDS(1) (I9) Node number.	NDS(2) (I9) Node number.	NOTE: Node numbers can be repeated for up to a maximum of 20.					S (3.6.15)	
Job Description, Format 18A4								IT1 (3.6.16)	

Columns 1-10	Columns 11-15	Columns 16-20	Columns 11-75	Columns 76-80
TIM(D10.0) Initial time.	INITEM (I9) Total number of lattice points with explicitly-specified initial temperatures. Maximum 1750	INITX1 (I9) Total number of lattice points with explicitly specified initial melting ratios. Maximum 1750	NOTE: INITEM specifies number of points entered on the IT3 cards. NOTE: INITX1 specified number of IT4 cards.	IT2 (3.6.16)

Columns 1-5	Columns 6-15	Columns 16-20	Columns 21-30	Columns 31-35	Columns 36-45	Columns 46-50	Columns 51-60	Columns 61-65	Columns 66-75	Columns 76-80
N (I5) Lattice point number.	T1(N) (D10.0) Specified initial temperature of that point.	N (I5) Lattice point number.	T1(N) (D10.0) Specified initial temperature of that point.	Can be repeated up to 1750 lattice points.						IT3 (3.6.16)

Columns 1-5	Columns 6-15	Columns 16-25	Columns 76-80
			Columns 76-80
N (I5) Lattice point number.	MELMAT(N) (I5) Material number associated with node N which is currently changing phase.	X1 (N) (D10.0) Specified initial melting ratio for point N for material MELMAT(N).	IT4 (3.6.16)

Columns 1-9	Columns 10-18	Columns 19-27	Columns 28-36	Columns 37-45	Columns 46-54	Columns 55-63	Columns 64-72	Columns 73-80
RESDUL (D9.0) Convergence criterion for implicit solution of transient equation. Corresponds to ε_1 in Eq. (2.37). (Default, 10^{-5})	REDUCR (D9.0) Convergence criterion for implicit solution of transient equation. Corresponds to ε_2 in Eq. (2.38). (Inactive)	ABSDIF (D9.0) Convergence criterion for implicit solution of transient equation involving temperature-dependent properties. Corresponds to ε_3 in Eq.(2.41). (Default 10^{-5})	THETA (D9.0) Parameter defining different techniques for transient equation. $0.5 \leq$ THETA ≤ 1.0. 0.5 implies Crank-Nicolson 1.0 implies Backwards Euler. Corresponds to θ in Eq. (2.34). (Default, 0.5)	BETAT (D9.0) Initial value for SOR acceleration parameter. Corresponds to ω in Eq. (2.34). If BETAT = 0, it will be optimized empirically. If BETAT < 0, it will be optimized using Carre's technique. If BETAT > 0, it will be constant. $0 < \lvert$BETAT$\rvert < 2$	NUPBTA (I9) Number of time steps between attempted acceleration parameter updates. Used when BETAT = 0. Corresponds to N_ω in Sec. 2.3.3(d). (Default, 1)	ITLRCO (I9) For BETAT = 0, Number-of-iterations criterion which initiates acceleration parameter updates. Corresponds to I_ω in Sec. 2.3.3(d). (Default, 5) For BETAT < 0, Number of iterations between SOR acceleration parameter updates. (Default, 12)	ITLRCI (I9) Number-of-iterations criterion to terminate acceleration parameter updates. Corresponds to J_ω in Sec. 2.3.3(d) (Default, 2)	IP (3.6.17)
DELTAT (D9.0) Initial time step for implicit solution of transient equation.	TSFACT (D9.0) Factor by which the current time step is multiplied at each time step. (Default, 1.0)	TSMAX (D9.0) Maximum size of time step. (Default, 10^{50})	TSCHGE (D9.0) Maximum time for which this series of time steps applies. When the current time exceeds TSCHGE, another TP card is read. (Default, 10^{50})	TPCGMX (D9.0) Maximum temperature charge allowed at a node over a time step.	PTPCGM (D9.0) Maximum percentage of relative change in temperature allowed at a node over a time step.	TSMIN (D9.0) Minimum size of time step. (Default, DELTAT/10)		TP (3.6.17)
Blank card if additional problem follows.								(3.6.18)
Cards for additional problem if desired.								

4. OUTPUT DESCRIPTION

4.1. General

The HEATING5 Program automatically lists the input data and prints out the initial temperature distribution for any type of problem. The final temperature distribution is printed for steady-state problems, and the temperatures are presented as a function of time for transient calculations. As indicated in Section 3.6.4(f), the final temperature distribution can be written on a specified unit (e.g., the card punch). Information can also be written on a specified unit and used by a plotting package to create various types of plots [see Section 3.6.4(b)].

4.2. Input Return

All of the input data are recapitulated in the output, appearing before the temperature distributions. The format is best illustrated by example, and the reader is referred to the sample problem. A table indicating the nodal connections for surface-to-surface heat transfer is also included. A table containing each node and its corresponding stability criterion is presented for transient problems.

4.3. Temperature Map

The temperatures for nodes in each plane are printed out in the form of a two-dimensional map. For three-dimensional problems, the temperature output map is presented for each XY or Rθ plane. However, one can specify that the map be presented for each XZ or RZ plane instead. See Section 3.6.3(e) for details. The indices of both the gross grid lines and the fine grid lines as well as the actual value of the fine grid lines are listed along the top for the horizontal direction and along the left for the vertical direction. The temperatures are then tabulated in a two-dimensional array. A matrix composed of the symbols "–" and "I" for the horizontal and vertical directions, respectively, is superimposed over the two-dimensional temperature array. The symbols are used as a divider between materials. Thus, the user can quickly and easily identify temperatures anywhere in the mesh. To facilitate the identification of the temperature of each node, a map of the node numbers in each plane is printed out in the same format immediately before the initial temperature distribution is depicted. The basic features of this output option were taken from the package of heat transfer codes, ORTHIS/ORTHAT (Ref. 9).

4.4. Steady-State Temperature Distribution

For the steady-state calculations, the program writes "STEADY STATE TEMPERATURE DISTRIBUTION" and indicates the number of steady-state iterations completed and the time at which the steady-state calculations were performed. The temperature distribution is then presented in map form as discussed above. The maximum and minimum temperatures and the nodes at which they occur are also listed. The boundary temperatures are then listed in a table along with a message indicating the status of the calculation. The elapsed computer time, measured in seconds from the start of the job, is also printed out.

4.5. Transient Temperature Distribution

For the transient calculation, the program writes "TRANSIENT TEMPERATURE DISTRIBUTION" and indicates the number of time steps completed for each specified time and the time at which the temperature distribution is written. The temperature distribution is listed in the same manner as for steady-state problems.

4.6. Map of Melting Ratios for Change-of-Phase Calculations

For problems involving change of phase, a map of melting ratios is printed out after each transient temperature distribution. The melting ratios for nodes in each plane are presented in a map similar to the one for the temperature distribution. The melting ratios for materials which cannot undergo a phase change are set to 1.0×10^{10} which causes the value to be printed out as asterisks on an IBM 360 system. Thus, it is easy to locate those nodes which cannot change phase, i.e., none of the materials associated with them are allowed to undergo a change of phase. If a node is associated with a material which can change phase, then the melting ratio as described in Section 2.6 is printed out.

4.7. Special Monitoring of Temperatures

One may wish to follow the temperatures of a few of the nodes during the calculations. As optional output, one may tabulate temperatures of specified nodes as a function of the number of iterations for steady-state problems or the number of time steps for transient problems. For more details on this option, see Section 3.6.4(d).

ACKNOWLEDGMENTS

This work is the result of an intermittent effort which has extended over a period of several years and has been supported at various stages by a number of projects or divisions in Oak Ridge. These have included Engineering Technology's High Temperature Gas-Cooled Reactor Safety Studies program directed by J. P. Sanders and funded by the Regulatory Segment of the U. S. Nuclear Regulatory Commission; the Office of Waste Isolation; and various projects within the Divisions of Computer Sciences, Engineering, Chemical Technology, Engineering Technology, Fusion Energy, and Metals and Ceramics. Work on an earlier version of this code was funded by the Transportation Branch, Division of Materials and Fuel Cycle Facility Licensing, U. S. Nuclear Regulatory Commission.

Special thanks are extended to Professor Julius Smith of the University of Tennessee for his many valuable suggestions concerning improvements in the numerical techniques which have contributed significantly to the success of the HEATING5 code.

Although it would be impossible to list each person who has contributed to this work, the following people deserve special mention by having served as "guinea pigs" by volunteering to run problems on the code while it was in the developmental stages, thus providing valuable assistance in debugging the code on real problems: H. C. Claiborne and J. W. Wachter of the Chemical Technology Division, R. A. Just and G. H. Llewellyn of the Engineering Division, H. T. Yeh of the Fusion Energy Division, and G. E. Giles of the Computer Sciences Division. Many of the above-mentioned individuals have also made suggestions which have been incorporated into the code.

The authors also gratefully acknowledge the contributions of the following people to the successful completion of this project:

R. C. Durfee for the use of his subroutine to print out the temperature distribution in map form,

G. E. Giles and J. C. Cleveland for their constructive review of the manuscript,

R. D. Cheverton, J. P. Sanders and M. Siman-Tov who provided suggestions concerning features for the code, and

K. T. Barry and J. V. Hughes for patiently retyping the manuscript at various stages of revision, and M. H. Cuthbert for editorial suggestions and the final preparation of this report.

REFERENCES

1. R. R. Liguori and J. W. Stephenson, *The HEATING Program*, ASTRA, Inc., Raleigh, North Carolina, January, 1961 (ASTRA 417-5.0).

2. T. B. Fowler and E. R. Volk, *Generalized Heat Conduction Code for the IBM-704 Computer,* Oak Ridge National Laboratory, Oak Ridge, Tennessee, October, 1959 (ORNL-2734).

3. W. D. Turner and J. S. Crowell, *Notes on HEATING – An IBM 360 Heat Conduction Program*, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, November, 1969 (CTC/INF-980).

4. W. D. Turner and M. Siman-Tov, *HEATING3 – An IBM 360 Heat Conduction Program*, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, February, 1971 (ORNL/TM-3208).

5. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Incorporated, Englewood Cliffs, New Jersey, 1962.

6. S. K. Hellman, G. Habetler, and H. Babrov, *Use of Numerical Analysis in the Transient Solution of Two-Dimensional Heat-Transfer Problem with Natural and Forced Convection*, Trans. Am. Soc. Mech. Engrs., 78, pp. 1155-61, 1956.

7. S. Levy, *Use of "Explicit Method" in Heat Transfer Calculations with an Arbitrary Time Step*, General Electric, Schenectady, New York, August, 1968 (GE Report No. 68-C-282).

8. B. A. Carré, *The Determination of the Optimum Accelerating Factor for Successive Over-Relaxation*, Comp. J., Vol. 4, pp. 73-78, 1961.

9. R. C. Durfee and C. W. Nestor, Jr., *ORTHIS/ORTHAT – Two Computer Programs for Solving Two-Dimensional Steady-State and Transient Heat Conduction Problems*, Union Carbide Corporation, Nuclear Division, Oak Ridge, Tennessee, July, 1971 (ORNL/TM-3224).

APPENDICES

APPENDIX A

CONTROL CARDS

A version of HEATING5, which can be used to solve problems containing up to 1400 nodes, is stored on disks at the computing centers at ORNL and ORGDP. This appendix discusses the Job Control Language (JCL) which is required to run this version of HEATING5 along with modifications to the JCL which are necessary for some of the optional uses of the code. This information will allow the user to solve most of his problems on the code. For additional information concerning JCL, the user is referred to the Oak Ridge Programmer's Notebook.

The control cards, which are necessary to use this version of HEATING5 at ORNL and ORGDP are presented in Figures A-1 and A-2, respectively. The first and second control cards are called the JOB statement and CLASS statement, respectively. The job name has a format of UIDXX where UID is the userid assigned to the user by the Computer Sciences Division (CSD) and the characters XX are any characters which make the job name unique. The accounting field (cccc) contains a four- or five-digit CSD charge number. The programmer-name field contains information necessary to route the job output back to the user. Since the default on core size will be exceeded for this version of HEATING5, a CLASS card must be included. See the Oak Ridge Programmer's Notebook for additional information concerning the JOB and CLASS statements.

If a request to save the final temperature distribution is made by inserting a unit number in the seventh entry on Card 4 (see Section 3.6.4(f)), then a DD card describing the output unit must appear before the

```
//GO.FT05F001 DD *
```

card. The following card

```
//GO.FT07F001 DD SYSOUT=B
```

would be a typical DD card for punching the final temperature distribution on cards on unit 7. The integer 7 would be entered in Entry 7 of Card 4 to define the unit number. If a unit number other than 7 is specified, then the DCB parameters must be specified on the DD card.

If one wishes to save the output temperature distributions in order to generate various plots with a plotting package, then a DD card describing the output data set must be inserted just before the

```
//GO.FT05F001 DD *
```

card. If a data set involving 450 nodes and two boundary conditions is to be saved on tape, then a typical DD card might be

```
//GO.FT08F001 DD UNIT=TAPE9,DISP=OLD,VOL=SER=08,LABEL=(,NL),
//  DCB=(RECFM=VBS,LRECL=3620,BLKSIZE=3624)
```

where 8 is the unit number which must be inserted in Entry 3 of Card 3. The SPECIAL=TAPE parameter must be added to the CLASS card for a job which is to be run at ORNL.

Two trivial modifications in HEATING5 will allow the code to solve problems containing up to a maximum of 6000 nodes. The main program for HEATING5 is shown in Fig. A-3. Let N be the maximum number of nodes which are desired for this case. The variable N must

be initialized in the INTEGER statement, and the variable CORE must be dimensioned 26 times N. The amount of core in K bytes required by HEATING5 is given roughly by the equation

$$M = (250 + 0.208N) . \tag{A-1}$$

Thus, the code can be easily modified to fit into a specified amount of core or to solve a problem with a specified number of nodes. However, when the number of nodes exceeds 1400, the REGION parameter on the CLASS card and the REGION.GO parameter on the EXEC card must be changed accordingly. The format of the JCL which is required to run HEATING5 with the main program modified is presented in Fig. A-4. The modified MAIN routine is inserted in the location indicated by the statement "(Enter FORTRAN Deck or Decks Here)."

Estimates for the CPU time (in seconds) which is required for problems run on the IBM 360/91 are given by Eqs. (A-2) and (A-3) for steady-state and explicit transient calculations, respectively.

$$(CPU)_{ss} = (1.0 \times 10^{-4})(\text{No. of Nodes})(\text{No. of Iterations}) \tag{A-2}$$

$$(CPU)_{t} = (5.0 \times 10^{-5})(\text{No. of Nodes})(\text{No. of Time Steps}) \tag{A-3}$$

For the implicit transient algorithm, the amount of CPU time per time step will be much larger than that for one of the explicit techniques since the code must do a lot more computation. However, one should be able to use a time step much larger than would be required in the explicit technique, especially for later times in the transient. Frequently, the implicit algorithm is more than an order of magnitude faster than the explicit algorithm in solving a problem. Temperature-dependent thermal properties and/or boundary conditions can increase the running time by up to an order of magnitude depending on how many nodes are associated with the temperature-dependent parameters. Equations (A-2) and (A-3) were derived empirically by examining a large number of cases which were run on the HEATING4 code, an earlier version of HEATING5. It is emphasized that these equations are merely estimates, and the actual CPU time may be considerably different from the estimate depending on the options which are used.

```
//jobname JOB (ccccc),'programmer-name'
//*CLASS REGION=540K,CPUxx=yyyz,LINES=lll
//  EXEC FORTHLG,REGION.GO=540K,PARM.GO='EU=-1'
//LKED.HEATING5 DD DISP=SHR,DSN=ONLINEA.WDTHF961.HEATING5
//LKED.SYSIN DD *
 INCLUDE HEATING5
/*
//GO.FT05F001 DD *
(Insert Data Deck Here)
/*
//
```

Fig. A-1. Format of JCL to Run HEATING5 at ORNL

```
//jobname JOB (cccc),'programmer-name'
//*CLASS REGION=540K,CPU95=yyyz,LINES=lll
//  EXEC FORTHLG,REGION.GO=540K,PARM.GO='EU=-1'
//LKED.HEATING5 DD DISP=SHR,DSN=A.WDTHF961.HEATING5
//LKED.SYSIN DD *
 INCLUDE HEATING5
/*
//GO.FT05F001 DD *
(Insert Data Decks Here)
/*
//
```

Fig. A-2. Format of JCL to Run HEATING5 at ORGDP

```
C ===============================================================
C ===== HEATING5 ================================================
C ===============================================================
C ===== THE HEATING CODE WITH TEMPERATURE-DEPENDENT THERMAL PROPERTIES,
C ===== NON-LINEAR AND SURFACE-TO-SURFACE BOUNDARY CONDITIONS,
C ===== CHANGE-OF-PHASE CAPABILITIES, AND AN IMPLICIT TECHNIQUE FOR
C ===== THE SOLUTION OF TRANSIENT PROBLEMS.
C ===============================================================
C ===== DEVELOPED BY W.D. TURNER,
C =====              D.C. ELROD,
C =====              I.I. SIMAN-TOV,
C ===== UNION CARBIDE CORPORATION, NUCLEAR DIVISION,
C ===== P.O. BOX Y,
C ===== OAK RIDGE, TN 37830,
C ===== RCOM 17, BLDG 9104-3,
C ===== PHONE 615-483-8611, EXT 3-5641 OR 3-3101.
C ===============================================================
C*****************************************************************
C*       LET MAXPTS = MAXIMUM NUMBER OF LATTICE POINTS.
C*       IF ONE WISHES TO CHANGE THE MAXIMUM NUMBER OF LATTICE POINTS,
C*       THEN THE FOLLOWING TWO STATEMENTS MUST BE MODIFIED AS
C*       INDICATED.
C*****************************************************************
C*     INTEGER N/MAXPTS/
C*     REAL*8 CORE(26MAXPTS)
      INTEGER  N/100/
      REAL*8 CORE(2600)
C*****************************************************************
      INTEGER*2 ICORE(1)
      LOGICAL*1 LCORE(1)
      EQUIVALENCE (CORE(1),ICORE(1),LCORE(1))
      CALL HEATN5(CORE,ICORE,LCORE,N)
      STOP
      END
```

Fig. A-3. Main Program for HEATING5, 100 Nodes

```
//jobname JOB (ccccc),'programmer-name'
//*CLASS REGION=540K,CPUxx=yyyz,LINES=lll
//  EXEC FORTHCLG,REGION.GO=540K,PARM.GO='EU=-1'
//FORT.SYSIN DD *
(Enter FORTRAN Deck or Decks Here)
/*
//LKED.HEATING5 DD DISP=SHR,DSN=ONLINEA.WDTHF961.HEATING5
//LKED.SYSIN DD *
 INCLUDE HEATING5
/*
//GO.FT05F001 DD *
(Insert Data Deck Here)
/*
//
```

Fig. A-4. Format of JCL to Run HEATING5 at ORNL when FORTRAN is Modified

APPENDIX B

NOMENCLATURE

Symbol	Description	Typical Units
h	Effective heat transfer coefficient	Btu/(hr·ft^2·°F)
h_c	Film coefficient (forced convection)	Btu/(hr·ft^2·°F)
h_e	Exponent for natural convection	Unitless
h_f	Heat flux	Btu/(hr·ft^2)
h_n	Coefficient for natural convection	Btu/(hr·ft^2·°F$^{(1+h_e)}$)
h_r	Coefficient for radiation	Btu/(hr·ft^2·°R^4)
k	Thermal conductivity	Btu/(hr·ft·°F)
Δq_i	Net heat flow into node i	Btu
r	Length along R-axis	ft
t	Time	hr
Δt	Time increment	hr
x	Length along X-axis	ft
y	Length along Y-axis	ft
z	Length along Z-axis	ft
A	Cross-sectional area normal to heat flow path	ft^2
$A_{m,i}$ or A_i	The i^{th} coefficient for the m^{th} analytical function	
B	Extrapolation factor used in steady-state algorithm	Unitless
C_i	Thermal capacitance of node i	Btu/°F
C_p	Specific heat	Btu/(lb·°F)
D_i	See Eq. (2.26)	
F(v)	Analytical function	
G(v)	Tabular function	
H_i	See Eq. (2.25)	
H_m	Latent heat of material m	Btu/lb
I	Total number of nodes whose temperatures must be calculated	
${}_iK_j$	Effective thermal conductance between nodes i and j	Btu/(hr·°F)

NOMENCLATURE (Contd.)

Symbol	Description	Typical Units
L	Length of heat flow path between two nodes	ft
L_i	Number of nodes associated with node i whose number is less than i	
M_i	Number of nodes connected to node i	
P_i	Power associated with node i	Btu/hr
P_o	Constant factor in definition of input parameter (see Eq. (3.4))	
Q	Power density	$Btu/(hr \cdot ft^3)$
R_i	Residual at node i in implicit transient algorithm (see Eq. (2.35))	Btu/hr
T_i	Temperature of node i	°F
T_{melt}	Transition temperature for a material	°F
V	Volume	ft^3
$X1_i$	Melting ratio associated with node i	Unitless
Z	Factor used in Levy's modification for transient algorithm (see Eq. (2.22))	Unitless
α_m	Number of m^{th} node connected to node i	
β	Acceleration parameter for SOR iteration method for steady-state algorithm	Unitless
ϵ	Convergence criterion for SOR iteration method for steady-state algorithm (see Eq. (2.14))	Unitless
ϵ_1	Convergence criterion for SOR iteration method for implicit transient algorithm (see Eq. (2.37))	Unitless
ϵ_2	Convergence criterion for SOR iteration method for implicit transient algorithm (see Eq. (2.38))	Unitless
ϵ_3	Convergence criterion for temperature-dependent iteration scheme for implicit transient algorithm (see Eq. (2.41))	Unitless
ρ	Density	lb/ft^3
θ	Length along θ-axis	Radians
Θ	Parameter in implicit algorithm for transient problems	Unitless
ω	Acceleration parameter for SOR iteration method for implicit transient algorithm	Unitless

NOMENCLATURE (Contd.)

Symbol	Description	Typical Units
Subscripts		
b	Boundary index	
i	The node under consideration	
j	A neighbor node connected to node i	
Superscripts		
m	Iteration number	
n	Time step number	
(n)	Iteration number	

APPENDIX C

USER-SUPPLIED SUBROUTINES

Subroutines may be supplied by the user to evaluate any of the parameters listed in Table 3.1. Thus, if an input parameter cannot be defined by one of the built-in functions as described in Section 3.5, it is quite simple for the user to add his own computational technique for evaluating the parameter. The user-supplied subroutine is referenced by specifying the parameter as an analytical function and by specifying no coefficients for the corresponding analytical function (i.e., leave the second entry on the A1 card blank and omit the related A2 card). Since this analytical function is only a flag to tell the code to call the appropriate user-supplied subroutine, the same analytical function can be specified for more than one parameter. The computational technique is then programmed in the subroutine associated with the parameter of interest (see Table 3.1), and the job is submitted to the computer according to the format presented in Fig. A-4.

HEATING5 contains dummy subroutines for each of the parameters listed in Table 3.1. If the user references one of the routines but fails to supply his own, then the code will write out an error message and stop when that subroutine is called. Each user-supplied subroutine has the same argument list. However, all five independent variables are not initialized for each subroutine. Only the independent variables marked by an x in Table 3.1 are initialized when each respective subroutine is called. Subroutine BNDTMP, a typical, dummy, user-supplied subroutine, is shown in Fig. C-1. With the exception of HEATGN, all the other user-supplied subroutines are basically the same as BNDTMP, the only differences being the variables which are initialized (see Table 3.1) and the parameter which is being evaluated. Subroutine HEATGN is shown in Fig. C-2.

If the thermal conductivity of a material is anisotropic, then it must be defined in user-supplied CONDTN (see Section 3.6.7).

```
      SUBROUTINE BNDTMP(RVALUE,R,TH,Z,TIM,TSN,VALUE,NUMBER,N)
C ***************************************************************************
C
C WHEN IT IS REFERENCED, THIS SUBROUTINE MUST BE SUPPLIED BY THE USER TO
C EVALUATE THE USER-SUPPLIED FUNCTION DEFINING THE BOUNDARY TEMPERATURE.
C THIS PARAMETER MAY BE TIME-DEPENDENT.
C THE VARIABLES IN THE ARGUMENT LIST ARE DEFINED IN THE FOLLOWING TABLE.
C (Y) INDICATES THAT THE VARIABLE HAS BEEN DEFINED WHEN THIS SUBROUTINE
C IS CALLED.  (N) INDICATES IT HAS NOT BEEN DEFINED.
C
C     VARIABLE    TYPE        DEFINITION
C     RVALUE      REAL*8      VALUE OF THE USER-SUPPLIED FUNCTION
C                             EVALUATING THE BOUNDARY TEMPERATURE (THIS
C                             VALUE MUST BE COMPUTED BY THIS ROUTINE). (N)
C                             NAME OF VARIABLE IN CALLING SEQUENCE --
C                             TDUM(I).
C     R           REAL*8      X OR R COORDINATE OF NODE N. (N)
C     TH          REAL*8      Y OR THETA COORDINATE OF NODE N. (N)
C     Z           REAL*8      Z COORDINATE OF NODE N. (N)
C     TIM         REAL*8      TIME AT WHICH PARAMETER IS TO BE
C                             EVALUATED. (Y)
C     TSN         REAL*8      TEMPERATURE AT WHICH PARAMETER IS TO BE
C                             EVALUATED. (N)
C     VALUE       REAL*8      CONSTANT VALUE OF THE PARAMETER WHICH
C                             APPEARS ON ITS RESPECTIVE INPUT CARD (B1) IF
C                             IT IS NON-ZERO.
C                             IF IT IS ZERO, VALUE CONTAINS 1.0. (Y)
C                             NAME OF VARIABLE IN CALLING SEQUENCE --
C                             BYTEMP(NBDTP).
C     NUMBER      INTEGER*4   INDEX OF PARAMETER BEING EVALUATED. (Y)
C                             NAME OF VARIABLE IN CALLING SEQUENCE - NBDTP
C     N           INTEGER*4   NODE NUMBER. (N)
C
C ***************************************************************************
      REAL*8 RVALUE,R,TH,Z,TIM,TSN,VALUE
      COMMON /INOUT/  IN    , IO
C
C INSERT ALGORITHM TO COMPUTE THE BOUNDARY TEMPERATURE FOR BOUNDARY
C CONDITION 'NUMBER' HERE.
C
      WRITE(IO,9000)
 9000 FORMAT('1***********'/
     1 ' YOU HAVE CALLED SUBROUTINE BNDTMP WHICH IS A USER-SUPPLIED FUNC
     2TION.'/
     3 ' HOWEVER, YOU HAVE NOT SUPPLIED THIS SUBROUTINE.'/
     4 ' EITHER SUPPLY SUBROUTINE BNDTMP OR CORRECT THE INPUT DATA SO TH
     5AT THIS SUBROUTINE IS NOT REFERENCED.')
      STOP
      END
```

Fig. C-1. Dummy User-Supplied Subroutine for Boundary Temperature

```
      SUBROUTINE HEATGN(RVALUE,R,TH,Z,TIM,TSN,VALUE,NUMBER,N)
C ***************************************************************************
C
C WHEN IT IS REFERENCED, THIS SUBROUTINE MUST BE SUPPLIED BY THE USER TO
C EVALUATE THE USER-SUPPLIED FUNCTION DEFINING THE VOLUMETRIC HEAT
C GENERATION RATE.  THIS PARAMETER MAY BE TIME-, TEMPERATURE- AND/OR
C POSITION-DEPENDENT.
C
C    NOTE: IF THE HEAT GENERATION RATE IS WRITTEN AS
C             Q = F(R,TH,Z,TSN) * G(TIM)
C          THEN THIS SUBROUTINE MAY BE CALLED FIRST TO CALCULATE THE
C          FACTOR F.  THEN, THIS ROUTINE IS CALLED FROM THE SOLUTION
C          ALGORITHM TO UPDATE THE TIME-DEPENDENT FACTOR G OF
C          THE HEAT GENERATION RATE.  WHEN THIS FACTOR IS UPDATED, ONLY
C          TIM, NUMBER, AND VALUE ARE DEFINED WHERE VALUE = 1.0.
C          IF NECESSARY, THE TWO CALLS MAY BE IDENTIFIED BY THE VALUE OF
C          N.  IN THE FIRST CALL, N CONTAINS THE NUMBER OF THE NODE
C          WHOSE HEAT GENERATION RATE IS BEING EVALUATED.  IN THE SECOND
C          CALL, N IS SET TO ZERO.
C
C THE VARIABLES IN THE ARGUMENT LIST ARE DEFINED IN THE FOLLOWING TABLE.
C (Y) INDICATES THAT THE VARIABLE HAS BEEN DEFINED WHEN THIS SUBROUTINE
C IS CALLED.  (N) INDICATES IT HAS NOT BEEN DEFINED.
C
C     VARIABLE   TYPE        DEFINITION
C     RVALUE     REAL*8      VALUE OF THE USER-SUPPLIED FUNCTION
C                            EVALUATING THE VOLUMETRIC HEAT GENERATION
C                            RATE (THIS VALUE MUST BE COMPUTED BY THIS
C                            ROUTINE). (N)
C                            NAME OF VARIABLE IN CALLING SEQUENCE - GEN.
C     R          REAL*8      X OR R COORDINATE OF NODE N. (Y)
C     TH         REAL*8      Y OR THETA COORDINATE OF NODE N. (Y)
C     Z          REAL*8      Z COORDINATE OF NODE N. (Y)
C     TIM        REAL*8      TIME AT WHICH PARAMETER IS TO BE
C                            EVALUATED. (Y)
C     TSN        REAL*8      TEMPERATURE AT WHICH PARAMETER IS TO BE
C                            EVALUATED. (Y)
C     VALUE      REAL*8      CONSTANT VALUE OF THE PARAMETER WHICH
C                            APPEARS ON ITS RESPECTIVE INPUT CARD (G) IF
C                            IT IS NON-ZERO.
C                            IF IT IS ZERO, VALUE CONTAINS 1.0. (Y)
C                            NAME OF VARIABLE IN CALLING SEQUENCE --
C                            GEN0(NGN).
C     NUMBER     INTEGER*4   INDEX OF PARAMETER BEING EVALUATED. (Y)
C                            NAME OF VARIABLE IN CALLING SEQUENCE - NGN.
C     N          INTEGER*4   NODE NUMBER. (Y)
C
C ***************************************************************************
      REAL*8 RVALUE,R,TH,Z,TIM,TSN,VALUE
      COMMON /INOUT/  IN   , IO
```

Fig. C-2. Dummy User-Supplied Subroutine for Volumetric Heat Generation Rate

```
C
C INSERT ALGORITHM TO COMPUTE THE VOLUMETRIC HEAT GENERATION RATE HERE.
C
      WRITE(IO,9000)
 9000 FORMAT('1**********'/
     1 ' YOU HAVE CALLED SUBROUTINE HEATGN WHICH IS A USER-SUPPLIED FUNC
     2TION.'/
     3 ' HOWEVER, YOU HAVE NOT SUPPLIED THIS SUBROUTINE.'/
     4 ' EITHER SUPPLY SUBROUTINE HEATGN OR CORRECT THE INPUT DATA SO TH
     5AT THIS SUBROUTINE IS NOT REFERENCED.')
      STOP
      END
```

Fig. C-2 (Contd.)

APPENDIX D

SAMPLE PROBLEM

An illustrated example was chosen for instructive purposes rather than representing a real engineering problem. The problem was two-dimensional in X-Y coordinates and consisted of three materials. Its configuration is shown in Fig. D-1. Numbers in circles represent regions and numbers in square frames represent boundary conditions. The units used were Btu, °F, lb, inch, and min. The conditions of the problem were as follows:

Regions 1 to 6 consisted of material No. 1 (iron); regions 7 to 9 consisted of material No. 2 (stainless steel); and regions 10, 11, and 12 were an air gap between the two metals. The physical properties of these materials are given in Table D-1.

Table D-1. Material Physical Properties for Test Problem Number 3 for HEATING5

Property/ Material	Iron (Material No. 1)	Stainless (Material No. 2)	Air (Material No. 3)
Conductivity	0.0296 at 0° F	0.013 at 0° F	1.82×10^{-5} at 0° F
Btu/(min·in·° F)	0.0264 at 752° F	0.015 at 752° F	3.41×10^{-5} at 500° F
	0.0222 at 1832° F	0.025 at 1832° F	4.68×10^{-5} at 1000° F
			5.75×10^{-5} at 1500° F
Density	0.2801	0.2824	5.00×10^{-5} at 0° F
lb/in.^{3}			2.39×10^{-5} at 500° F
			1.57×10^{-5} at 1000° F
			1.17×10^{-5} at 1500° F
Specific Heat	0.116	0.11	0.25
Btu/(lb·° F)			

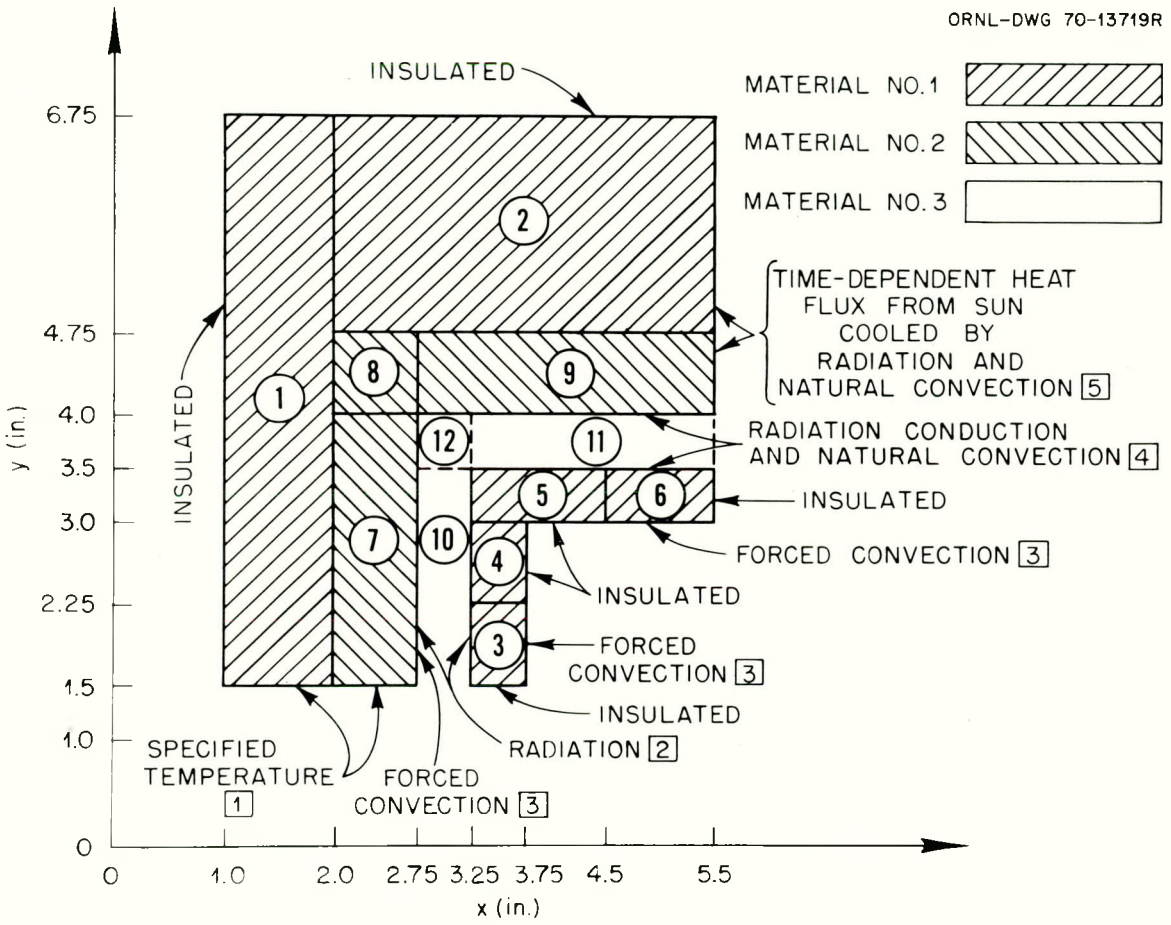

Fig. D-1 Two-Dimensional XY Test Problem Number 3 for HEATING5

There was a uniform heat generation in regions 1 and 2 at the rate of 1.0 Btu/(min·in.^{3}) which was time-dependent according to time function No. 2 given in Fig. D-2. The initial temperature all over was 100°F. The boundary conditions on each of the faces are shown in Fig. D-1, and they are numbered in square frames. Boundary condition No. 1 was in perfect contact with a fluid at 200°F and was time-dependent according to time function No. 1 given in Fig. D-2. Boundary condition No. 2 was radiation across an air gap (region 10) between the two metals (emissivity $\epsilon = 0.8$). Conduction and natural convection were neglected. Boundary condition No. 3 was forced convection to a fluid at 68°F (one face of region 10 only). The heat transfer coefficient was 0.006 Btu/(min·in^2·°F). Boundary Condition No. 4 was combined heat transfer by radiation and natural convection across an air gap (region 11) between the two metals (emissivity $\epsilon = 0.8$). Heat was also transferred by conduction through the air. The natural convective heat transfer coefficient was given by

$$h = 2.56 \times 10^{-5} \Delta T^{0.33} \text{ Btu/(min·in}^2\text{·°F)}. \tag{D-1}$$

Boundary condition No. 5 was a time-dependent heat flux from solar radiation given by Eq. (D-2) and was cooled by radiation and natural convection to the open atmosphere (at 100°F). The rest of the boundaries were insulated.

$$h_f = 0.03\cos\left(\frac{\pi}{360}t\right) \tag{D-2}$$

Region 12 could not be described for surface-to-surface radiation or natural convection because of lack of opposing surfaces. Conduction could have been taken into account but was neglected in this case. It was desired to know the transient temperature distribution at 30 and 60 minutes and the steady-state temperature distribution resulting from evaluating all time functions at 60 minutes. It was also desired to monitor the temperatures at points (1.0,1.5) (3.75,3.0), (2.75,4.0), (5.5,4.0) and (5.5,6.75) after every 10 time steps or iterations. The maximum CPU time was limited to 100 seconds.

The problem was run on HEATING5 using the input data given in Fig. D-3. The transient calculations used the Crank-Nicolson Procedure with an initial time step size of 0.1 minutes which was increased 10% after each time step. The printed input and output are given in Table D-2. The IBM 360/195 CPU time for the case was 10.1 seconds, of which the 60-minute transient calculations required about 7.4 seconds.

The following changes to the model were made to demonstrate some additional capabilities of HEATING5. Suppose that the initial temperature varied as a function of y according to the following expression

$$T_o(y) = 235 - 20y \tag{D-3}$$

and the heat generation rate in region 1 was a sum of exponentials defined by the expression

$$Q_1(t) = \sum_{i=1}^{3} C_{i1} e^{-\lambda_{1i} t} \tag{D-4}$$

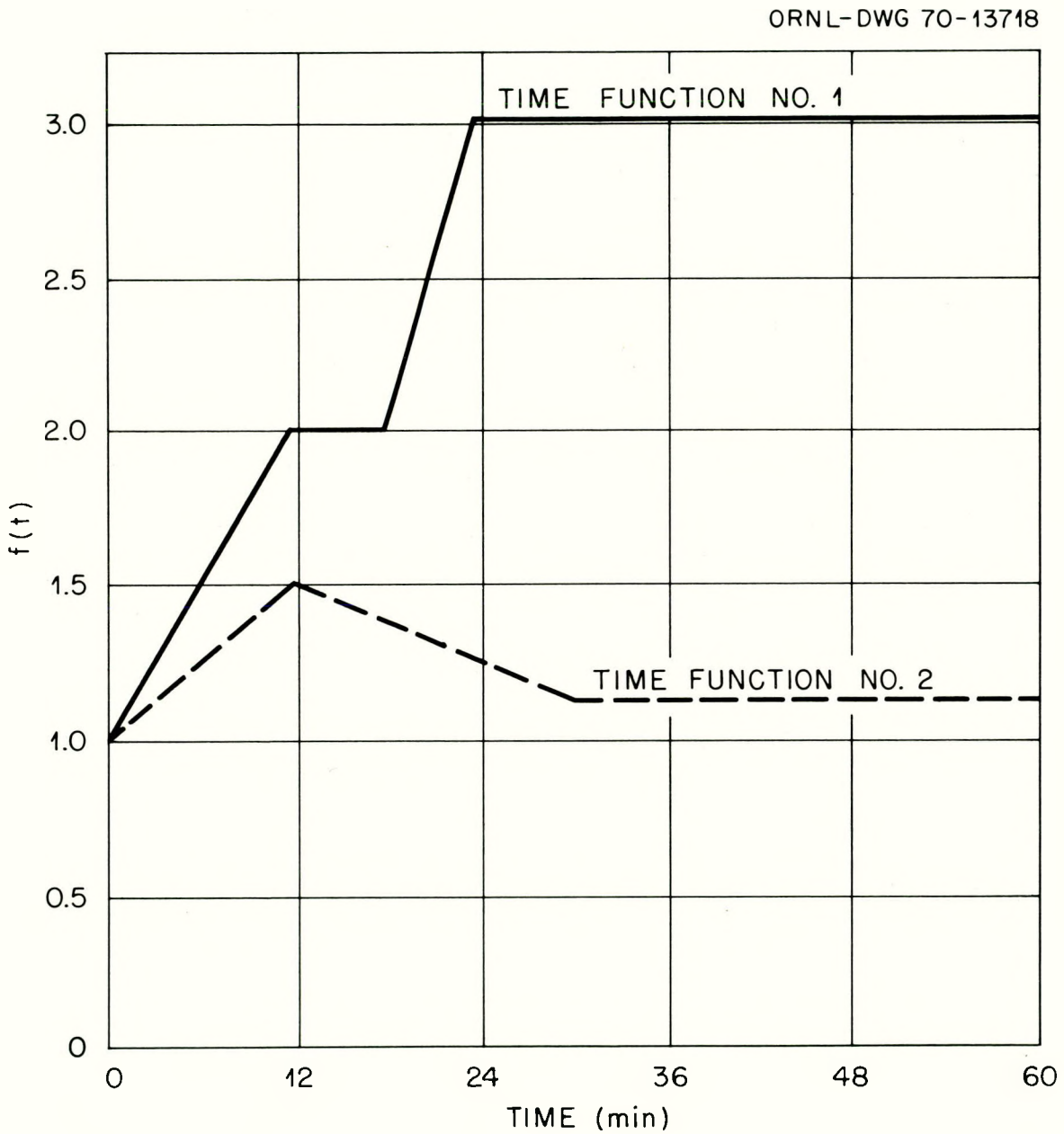

Fig. D-2 Time-Dependent Functions for Test Problem Number 3 for HEATING5

```
TEST PROBLEM #3 FOR HEATING5
     100         7        11         3                    1         1         5   CARD2
       7         7                   1         6                                  CARD3
                 2                             10                                 CARD4
      -2                                                                   60.0    CARD5
       1         1       1.0       2.0       1.5      6.75                        R1
       1         1                           1                                    R2
       2         1       2.0       5.5      4.75      6.75                        R1
       1         1                   5                                            R2
       3         1      3.25      3.75       1.5      2.25                        R1
       1                             3                                            R2
       4         1      3.25      3.75      2.25       3.0                        R1
       1                                                                         R2
       5         1      3.25       4.5       3.0       3.5                        R1
       1                                                                         R2
       6         1       4.5       5.5       3.0       3.5                        R1
       1                                       3                                 R2
       7         2       2.0      2.75       1.5       4.0                        R1
       1                             3         1                                 R2
       8         2       2.0      2.75       4.0      4.75                        R1
       1                                                                         R2
       9         2      2.75       5.5       4.0      4.75                        R1
       1                             5                                           R2
      10                 2.75      3.25       1.5       3.5                        R1
       1                    2         2                                          R2
      11         3      3.25       5.5       3.5       4.0                        R1
       1                                       4         4                       R2
       1 IRON                          .2801      .116        -3                  M
       2 STAINLSS                      .2824       .11        -4                  M
       3 AIR                                      0.25        -5        -6        M
       1       1.0        -2                                                     G
       1     100.0                                                              I
       1         2     200.0        -1                                          B1
                                                                                 B2
       2         3                                                               B1
         1.58D-13                                                                B2
       3         1      68.0                                                     B1
6.0D-3                                                                           B2
       4         3                                                               B1
         1.58D-13 2.56D-05      0.33                                             B2
       5         1     100.0                                                     B1
         1.58D-13 2.56D-05      0.33                    1                        B2
                                                     1                           B3
     1.0       2.0       2.75      3.25      3.75       4.5       5.5           L1
       2         1          1         1         1         1                       N1
     1.5      2.25       3.0       3.5       4.0      4.75      6.75           L2
       1         1          1         1         1         4                       N2
       1         2                                                             A1
       4      0.03          5 .0087266                                         A2
       1         4                                                             T1
```

Fig. D-3. Input Data for Test Problem Number 3 for HEATING5

```
     0.0        1.0       12.0        2.0       18.0        2.0       24.0        3.0      T2
       2          3                                                                        T1
     0.0        1.0       12.0        1.5       30.0      1.125                          T2
       3          3                                                                        T1
     0.0      .0296      752.0      .0264     1832.0      .0222                          T2
       4          3                                                                        T1
     0.0       .013      752.0      .0153     1832.0       .025                          T2
       5          4                                                                        T1
     0.0    1.82D-5      500.0    3.41D-5     1000.0    4.68D-5     1500.0    5.75D-5      T2
       6          4                                                                        T1
     0.0     5.0D-5      500.0    2.39D-5     1000.0    1.57D-5     1500.0    1.17D-5      T2
    30.0       60.0                                                                       D
       5          1         18          32          36          76                       S
1.0D-05    1.0D-3    1.0D-5                                                               IP
     0.1        1.1                                                                      TP
```

Fig. D-3 (Contd.)

where the parameters were defined as

i	C_{i1}	λ_{1i}
1	0.5	0.0115525
2	0.3	0.0231049
3	0.2	0.0462098

and the heat generation rate in region 2 was a sum of exponentials defined by the expression

$$Q_2(t) = \sum_{i=1}^{2} C_{i2} e^{-\lambda_{2i} t} \qquad \text{(D-5)}$$

where the parameters were defined as

i	C_{i2}	λ_{2i}
1	0.6	0.0115525
2	0.4	0.0462098

Furthermore, the thermal conductivity for iron was assumed to be anisotropic with the conductivity along the Y-axis equal to twice that along the X-axis as presented in Table D-1. The initial temperature was input to the code as an analytical function but the two heat generation rates and the conductivity for iron must be defined by user-supplied subroutines. The input data for this case is presented in Fig. D-4. Note that tabular function numbers 2 and 3 are part of the input data but are not used. The user-supplied subroutines for the heat generation rate and the thermal conductivity for iron are presented in Figs. D-5 and D-6, respectively.

Table D-2. Computer Output for Test Problem
Number 3 for HEATNG5

```
HEATING5. A MULTI-DIMENSIONAL HEAT CONDUCTION CODE WITH TEMPERATURE-DEPENDENT THERMAL PROPERTIES.
NON-LINEAR AND SURFACE-TO-SURFACE BOUNDARY CONDITIONS AND CHANGE-OF-PHASE CAPABILITIES.
THIS VERSION OF THE CODE IS DESCRIBED IN ORNL-TM-
THE TRANSIENT SOLUTION CAN BE CALCULATED BY AN IMPLICIT TECHNIQUE (CRANK-NICOLSON OR
BACKWARDS EULER) FOR PROBLEMS WITH MATERIALS WHICH ARE NOT ALLOWED TO UNDERGO A PHASE CHANGE.
THE ONE-DIMENSIONAL R SPHERICAL MODEL WAS ADDED NOV. 75. THIS MODEL MAY BE ACCESSED
BY SPECIFYING NGEOM = 10 IN THE INPUT DATA.
HEATING5 WAS WRITTEN BY

     W.D. TURNER
     D.C. ELROD
     I.I. SIMAN-TOV
     COMPUTER SCIENCES DIVISION
     UNION CARBIDE CORPORATION, NUCLEAR DIVISION
     OAK RIDGE, TENNESSEE 37830

THIS VERSION OF HEATING CAN HANDLE A MAXIMUM OF    100 LATTICE POINTS.

INPUT RETURN

  DATE 11-30-76
  TIME 14.04.04

  JOB DESCRIPTION--  TEST PROBLEM #3 FOR HEATING5
  THE PROBLEM WILL BE TERMINATED AFTER       100 SECONDS
  GEOMETRY TYPE NUMBER     7 (OR XY  )
  NUMBER OF REGIONS      11
  NUMBER OF MATERIALS      3
  NUMBER OF HEAT GENERATION FUNCTIONS     1
  NUMBER OF INITIAL TEMPERATURE FUNCTIONS       1
  NUMBER OF DIFFERENT KINDS OF BOUNDARIES     5
  THIS PROBLEM INVOLVES TEMPERATURE-DEPENDENT PROPERTIES.
  NUMBER OF POINTS IN GROSS X OR R LATTICE     7
  NUMBER OF POINTS IN GROSS Y OR THETA LATTICE     7
  NUMBER OF POINTS IN GROSS Z LATTICE     0
  NUMBER OF ANALYTIC FUNCTIONS     1
  NUMBER OF TABULAR FUNCTIONS     6
  SINCE THIS PROBLEM INVOLVES RADIATION, 459.69 WILL BE ADDED TO
  THE TEMPERATURES TO CONVERT THEM TO ABSOLUTE TEMPERATURES.
  NUMBER OF TRANSIENT PRINTOUTS SPECIFIED     2
  TEMPERATURES OF SELECTED NODES WILL BE MONITORED EVERY    10 ITERATIONS OR TIME STEPS.
  PROBLEM TYPE NUMBER    -2
  STEADY STATE CONVERGENCE CRITERION  1.0000000D-05
  MAXIMUM NUMBER OF STEADY-STATE ITERATIONS    500
  NUMBER OF ITERATIONS BETWEEN TEMPERATURE DEPENDENT
  PARAMETER EVALUATIONS FOR STEADY STATE CALCULATIONS      0
  INITIAL OVERRELAXATION FACTOR (BETA) FOR STEADY STATE CALCULATIONS  1.90000000
  TIME INCREMENT  0.0
  INITIAL TIME  0.0
  FINAL TIME  6.0000000D 01
```

Table D-2 (Contd.)

SUMMARY OF REGION DATA

NUMBERS AND FCN NUMBER				********** DIMENSIONS **********						-------- BOUNDARY NUMBERS --------					
REG. NO.	MATL NO.	INIT TEMP	HEAT GEN.	LEFT-X-OR INNER-R	RIGHT-X-OR OUTER-R	LOWER-Y-OR LEFT-THETA	UPPER-Y-OR RIGHT-THETA	REAR-Z	FRONT-Z	LF-X IN-R	RT-X OT-R	LO-Y LF-O	UP-Y RT-O	RR-Z	FT-Z
1	1	1	1	1.0000	2.0000	1.5000	6.7500	0.0	0.0	0	0	1	0	0	0
2	1	1	1	2.0000	5.5000	4.7500	6.7500	0.0	0.0	0	5	0	0	0	0
3	1	1	0	3.2500	3.7500	1.5000	2.2500	0.0	0.0	0	3	0	0	0	0
4	1	1	0	3.2500	3.7500	2.2500	3.0000	0.0	0.0	0	0	0	0	0	0
5	1	1	0	3.2500	4.5000	3.0000	3.5000	0.0	0.0	0	0	0	0	0	0
6	1	1	0	4.5000	5.5000	3.0000	3.5000	0.0	0.0	0	0	3	0	0	0
7	2	1	0	2.0000	2.7500	1.5000	4.0000	0.0	0.0	0	3	1	0	0	0
8	2	1	0	2.0000	2.7500	4.0000	4.7500	0.0	0.0	0	0	0	0	0	0
9	2	1	0	2.7500	5.5000	4.0000	4.7500	0.0	0.0	0	5	0	0	0	0
10	0	1	0	2.7500	3.2500	1.5000	3.5000	0.0	0.0	2	2	0	0	0	0
11	3	1	0	3.2500	5.5000	3.5000	4.0000	0.0	0.0	0	0	4	4	0	0

Table D-2 (Contd.)

********** SUMMARY OF MATERIAL DATA **********

MATERIAL NUMBER	MATERIAL NAME	THERMAL PARAMETERS -- TEMPERATURE-DEPENDENT FUNCTION NUMBERS -- CONDUCTIVITY	DENSITY	SPECIFIC HEAT
1	IRON	0.0	2.801000D-01	1.160000D-01
		-3	0	0
2	STAINLSS	0.0	2.824000D-01	1.100000D-01
		-4	0	0
3	AIR	0.0	0.0	2.500000D-01
		-5	-6	0

********** SUMMARY OF INITIAL TEMPERATURE DATA **********

NUMBER	INITIAL TEMPERATURE	POSITION-DEPENDENT FUNCTION NUMBERS X OR R	Y OR TH	Z
1	1.00000D 02	0	0	0

********** SUMMARY OF HEAT GENERATION RATE DATA **********

NUMBER	POWER DENSITY	TIME-, TEMPERATURE-, AND POSITION-DEPENDENT NUMBERS TIME	TEMPERATURE	X OR R	Y OR TH	Z
1	1.00000D 00	-2	0	0	0	0

Table D-2 (Contd.)

**********SUMMARY OF BOUNDARY DATA**********

-------GENERAL------			--TEMPERATURE-- INFORMATION	----------HEAT TRANSFER COEFFICIENTS---------- RELATED FUNCTION NUMBERS				
NO.	TYPE	FCT FLAG	TEMPERATURE & TIME FCT	ASSOC. FCTS	FORCED CONV.	RADIATION	NATURAL CONV	EXPONENT	FLUX
1	2	0	2.00000D 02 -1		0.0	0.0	0.0	0.0	0.0
2	3	0	0.0 0		0.0	1.58000D-13	0.0	0.0	0.0
3	1	0	6.80000D 01 0		6.00000D-03	0.0	0.0	0.0	0.0
4	3	0	0.0 0		0.0	1.58000D-13	2.56000D-05	3.30000D-01	0.0
5	1	1	1.00000D 02 0	TIME TEMP	0.0 0 0	1.58000D-13 0 0	2.56000D-05 0 0	3.30000D-01 0 0	0.0 1 0

GROSS LATTICES AND NUMBERS OF INCREMENTS

R OR X

1.000000	2.000000	2.750000	3.250000	3.750000	4.500000	5.500000
2	1	1	1	1	1	

THETA OR Y

1.500000	2.250000	3.000000	3.500000	4.000000	4.750000	6.750000
1	1	1	1	1	4	

LISTING OF ANALYTIC FUNCTIONS

F(V)= A(1) + A(2)*V + A(3)*V**2 + A(4)*COS(A(5)*V) + A(6)*EXP(A(7)*V) + A(8)*SIN(A(9)*V) + A(10)*LOG(A(11)*V)

NO.	A(1)	A(2)	A(3)	A(4)	A(5)	A(6)	A(7)	A(8)	A(9)	A(10)	A(11)
1	0.0	0.0	0.0	3.000D-02	8.727D-03	0.0	0.0	0.0	0.0	0.0	0.0

Table D-2 (Contd.)

```
                    LISTING  OF  TABULAR  FUNCTIONS

TABLE NUMBER        1          NUMBER OF PAIRS -        4

      ARGUMENT                       VALUE
    0.0                          1.00000000D 00
    1.20000000D 01               2.00000000D 00
    1.80000000D 01               2.00000000D 00
    2.40000000D 01               3.00000000D 00

TABLE NUMBER        2          NUMBER OF PAIRS -        3

      ARGUMENT                       VALUE
    0.0                          1.00000000D 00
    1.20000000D 01               1.50000000D 00
    3.00000000D 01               1.12500000D 00

TABLE NUMBER        3          NUMBER OF PAIRS -        3

      ARGUMENT                       VALUE
    0.0                          2.96000000D-02
    7.52000000D 02               2.64000000D-02
    1.83200000D 03               2.22000000D-02

TABLE NUMBER        4          NUMBER OF PAIRS -        3

      ARGUMENT                       VALUE
    0.0                          1.30000000D-02
    7.52000000D 02               1.53000000D-02
    1.83200000D 03               2.50000000D-02

TABLE NUMBER        5          NUMBER OF PAIRS -        4

      ARGUMENT                       VALUE
    0.0                          1.82000000D-05
    5.00000000D 02               3.41000000D-05
    1.00000000D 03               4.68000000D-05
    1.50000000D 03               5.75000000D-05

TABLE NUMBER        6          NUMBER OF PAIRS -        4

      ARGUMENT                       VALUE
    0.0                          5.00000000D-05
    5.00000000D 02               2.39000000D-05
    1.00000000D 03               1.57000000D-05
    1.50000000D 03               1.17000000D-05

TABLE   OF   OUTPUT   TIMES

OUTPUT        OUTPUT          OUTPUT        OUTPUT          OUTPUT        OUTPUT          OUTPUT        OUTPUT
NO.           TIME            NO.           TIME            NO.           TIME            NO.           TIME
    1       3.00000D 01
    2       6.00000D 01

TEMPERATURES OF THE FOLLOWING NODES WILL BE MONITORED
EVERY   10 ITERATIONS OR TIME STEPS.
NUMBER     NODE
   1          1
   2         18
   3         32
   4         36
   5         76
```

Table D-2 (Contd.)

```
FINE LATTICE, X OR R, Y OR THETA, AND Z

    1     1.000000      2     1.500000      3     2.000000      4     2.750000      5     3.250000
    6     3.750000      7     4.500000      8     5.500000

    1     1.500000      2     2.250000      3     3.000000      4     3.500000      5     4.000000
    6     4.750000      7     5.250000      8     5.750000      9     6.250000     10     6.750000

THIS PROBLEM CONTAINS    76 NODES.
```

Table D-2 (Contd.)

SURFACE-TO-SURFACE CONNECTIONS

NUMBER	NODE	TO	NODE
1	4		5
2	10		11
3	16		17
4	24		25
5	25		33
6	26		34
7	27		35
8	28		36

THE INITIAL TIME STEP = 1.000000D-01

AFTER EACH TIME STEP THE TIME STEP SIZE WILL BE MULTIPLIED BY A FACTOR OF 1.100000D 00 SUBJECT TO ANY CONDITIONS WHICH MAY FOLLOW.

THE MINIMUM TIME STEP ALLOWED IS 1.000000D-02

Table D-2 (Contd.)

STABILITY CRITERION FOR EACH NODE

1	0.0	2	0.0	3	0.0	4	0.0	5	9.6251D-02	6	8.9973D-02
7	9.6378D-02	8	9.6378D-02	9	1.6509D-01	10	2.8007D-01	11	9.6251D-02	12	9.3065D-02
13	8.3527D-02	14	8.3527D-02	15	1.3963D-01	16	2.3084D-01	17	8.3432D-02	18	8.7924D-02
19	1.0000D-01	20	1.0290D-01	21	6.9606D-02	22	6.9606D-02	23	1.1340D-01	24	1.8283D-01
25	6.9447D-02	26	8.3400D-02	27	1.0422D-01	28	1.1116D-01	29	8.3527D-02	30	8.3527D-02
31	1.3963D-01	32	2.3359D-01	33	2.0171D-01	34	2.6163D-01	35	3.7328D-01	36	4.1661D-01
37	8.3527D-02	38	8.3527D-02	39	1.1819D-01	40	1.3963D-01	41	1.1340D-01	42	1.3963D-01
43	1.8165D-01	44	1.9616D-01	45	6.9606D-02	46	6.9606D-02	47	8.3527D-02	48	8.3527D-02
49	6.9606D-02	50	8.3527D-02	51	1.0441D-01	52	1.1129D-01	53	6.9606D-02	54	6.9606D-02
55	8.3527D-02	56	8.3527D-02	57	6.9606D-02	58	8.3527D-02	59	1.0441D-01	60	1.1129D-01
61	6.9606D-02	62	6.9606D-02	63	8.3527D-02	64	8.3527D-02	65	6.9606D-02	66	8.3527D-02
67	1.0441D-01	68	1.1129D-01	69	6.9606D-02	70	6.9606D-02	71	8.3527D-02	72	8.3527D-02
73	6.9606D-02	74	8.3527D-02	75	1.0441D-01	76	1.1129D-01				

Table D-2 (Contd.)

```
THE STABILITY CRITERION IS  6.9447447D-02 FOR POINT    25
```

Table D-2 (Contd.)

MAP OF THE NODE NUMBERS

GROSS GRID			1		2	3	4	5	6	7
FINE GRID			1	2	3	4	5	6	7	8
		DISTANCE	1.00	1.50	2.00	2.75	3.25	3.75	4.50	5.50
1	1	1.50	1	2	3	4	5	6	0	0
2	2	2.25	7	8	9	10	11	12	0	0
3	3	3.00	13	14	15	16	17	18	19	20
4	4	3.50	21	22	23	24	25	26	27	28
5	5	4.00	29	30	31	32	33	34	35	36
6	6	4.75	37	38	39	40	41	42	43	44
	7	5.25	45	46	47	48	49	50	51	52
	8	5.75	53	54	55	56	57	58	59	60
	9	6.25	61	62	63	64	65	66	67	68
7	10	6.75	69	70	71	72	73	74	75	76

Table D-2 (Contd.)

TRANSIENT TEMPERATURE DISTRIBUTION AFTER 0 TIME STEPS, TIME = 0.0

GROSS GRID			1		2	3	4	5	6	7
			I		I	I	I	I	I	I
FINE GRID			1	2	3	4	5	6	7	8
		DISTANCE	1.00	1.50	2.00	2.75	3.25	3.75	4.50	5.50
1	1	1.50	100I00	+00.00	+00I00	+00I00	+00I00	+00I00	0.0	0I0
2	2	2.25	100I00	100.00	100I00	100I00	100I00	100I00	0.0	0I0
3	3	3.00	100I00	100.00	100I00	100I00	100I00	100+00	+00.00	+00I00
4	4	3.50	100I00	100.00	100I00	100I00	100+00	+00.00	+00.00	+00I00
5	5	4.00	100I00	100.00	100I00	100+00	+00.00	+00.00	+00.00	+00I00
6	6	4.75	100I00	100.00	100+00	+00.00	+00.00	+00.00	+00.00	+00I00
	7	5.25	100I00	100.00	100.00	100.00	100.00	100.00	100.00	100I00
	8	5.75	100I00	100.00	100.00	100.00	100.00	100.00	100.00	100I00
	9	6.25	100I00	100.00	100.00	100.00	100.00	100.00	100.00	100I00
7	10	6.75	100I00	+00.00	+00.00	+00.00	+00.00	+00.00	+00.00	+00I00

TEMPERATURES ON NUMBERED BOUNDARIES

BOUNDARY NUMBER	TEMPERATURE
1	200.000000
2	0.0
3	68.000000
4	0.0
5	100.000000

THE CURRENT TIME STEP (DELTAT) = 9.09090909D-02

ELAPSED CPU TIME IS 0.31 SECONDS

THE MAXIMUM TEMPERATURE IS - 1.00000D 02 (+-0.1)

MAX. TEMP. APPEARS AT NODES -

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50

THE MINIMUM TEMPERATURE IS - 1.00000D 02 (+-0.1)

MIN. TEMP. APPEARS AT NODES -

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50

Table D-2 (Contd.)

```
THE IMPLICIT PROCEDURE WILL BE USED TO CALCULATE THE TRANSIENT TEMPERATURE DISTRIBUTION.
MAXIMUM NORMALIZED HEAT RESIDUAL CONVERGENCE CRITERION = 1.00000D-05
(CORRESPONDS TO EPSILON SUB 1)
REDUCTION IN NORMALIZED HEAT RESIDUAL CONVERGENCE CRITERION = 1.00000D-03
(CORRESPONDS TO EPSILON SUB 2)
AVERAGE L1 NORM OF RELATIVE TEMPERATURE DIFFERENCE CONVERGENCE CRITERION
FOR TEMPERATURE DEPENDENT PROPERTIES = 1.00000D-05
(CORRESPONDS TO EPSILON SUB 3)
THETA (0.5 FOR CRANK-NICOLSON,1.0 FOR CLASSICAL IMPLICIT) = 5.00000D-01

THE SOR ACCELERATION PARAMETER (BETA) WILL BE OPTIMIZED EMPIRICALLY.
BETA = 1.00000D 00
A BETA UPDATE WILL BE ATTEMPTED EVERY     1 TIME STEPS.
NUMBER-OF-ITERATIONS TOLERANCE FOR BETA UPDATE CALCULATIONS, OUTER LOOP =     5
NUMBER-OF-ITERATIONS TOLERANCE FOR BETA UPDATE CALCULATIONS, INNER LOOP =     2
PER CENT CHANGE IN BETA UPDATE CALCULATIONS = 10
```

Table D-2 (Contd.)

THE FOLLOWING TABLE IS PRINTED OUT FOR INFORMATION PURPOSES DURING THE IMPLICIT TRANSIENT CALCULATIONS. A LINE IS PRINTED EACH TIME THE INNER LOOP CONVERGES. A LINE IS ALSO PRINTED AFTER THE VERY FIRST ITERATION FOR EACH TIME STEP. THUS, ONE CAN DETERMINE HOW MUCH THE MAXIMUM NORMALIZED HEAT RESIDUAL DECREASES DURING THE ITERATIVE PROCESS. ENTRIES IN EACH COLUMN ARE DESCRIBED BELOW:

NO TIME	-- NUMBER OF TIME STEPS.
TIME	-- TIME AT WHICH TEMPERATURE DISTRIBUTION IS BEING CALCULATED.
NO ITER	-- NUMBER OF ITERATIONS REQUIRED FOR INNER (LINEAR) LOOP TO CONVERGE.
MAX HEAT RESIDUAL	-- THE MAXIMUM NORMALIZED HEAT RESIDUAL AFTER THE NUMBER OF ITERATIONS INDICATED IN THE PREVIOUS COLUMN (COMPARES TO EPSILON SUB 1).
BETA	-- CURRENT VALUE OF THE SOR ACCELERATION PARAMETER.
L1 NORM OF TEMP DIFF	-- THE L1 NORM OF THE TEMPERATURE DIFFERENCE OVER THE CURRENT ITERATION FOR INNER (LINEAR) LOOP. THIS COLUMN AND THE NEXT TWO ARE USED ONLY WHEN THE OPTIMUM ACCELERATION PARAMETER IS BEING ESTIMATED USING CARRE'S TECHNIQUE.
RHO(ITERATION)	-- SPECTRAL RADIUS FOR THE SOR ITERATION MATRIX.
RHO(JACOBI)	-- SQUARE OF SPECTRAL RADIUS FOR THE JACOBI ITERATION MATRIX.
NO ITER	-- NUMBER OF ITERATIONS COMPLETED FOR OUTER (NON-LINEAR) LOOP.
L1 NORM OF TEMP DIFF	-- THE AVERAGE L1 NORM OF THE RELATIVE TEMPERATURE DIFFERENCE OVER THE CURRENT ITERATION FOR OUTER (NON-LINEAR) LOOP. NON-ZERO FOR NON-LINEAR PROBLEMS ONLY. (COMPARES TO EPSILON SUB 3)
NODE	-- NODE NUMBER.
MAX TEMP CHANGE	-- MAXIMUM TEMPERATURE CHANGE AT A NODE OVER THE CURRENT TIME STEP. THIS CHANGE OCCURRED AT THE NODE SHOWN IN THE PREVIOUS COLUMN.
NODE	-- NODE NUMBER.
MAX PERCENT TEMP CHANGE	-- MAXIMUM PERCENTAGE OF RELATIVE CHANGE IN TEMPERATURE AT A NODE OVER THE CURRENT TIME STEP. THIS CHANGE OCCURRED AT THE NODE SHOWN IN THE PREVIOUS COLUMN.

Table D-2 (Contd.)

NO TIME	TIME	NO ITER	MAX HEAT RESIDUAL	BETA	L1 NORM OF TEMP DIFF	RHO (ITERATION)	RHO (JACOBI)	NO ITER	L1 NORM OF TEMP DIFF	NODE	MAX TEMP CHANGE	NODE	MAX PERCENT TEMP CHANGE
1	1.00000D-01	0	5.59833D-03	1.00000D 00	0.0	0.0	0.0						
1	1.00000D-01	1	8.45128D-04	1.00000D 00	0.0	0.0	0.0						
1	1.00000D-01	5	2.39906D-06	1.00000D 00	0.0	0.0	0.0	1	1.21817D-02				
1	1.00000D-01	2	6.74186D-07	1.00000D 00	0.0	0.0	0.0	2	2.15646D-06	7	9.77621D 00	7	1.74672D 00
2	2.10000D-01	1	1.91541D-03	1.10000D 00	0.0	0.0	0.0						
2	2.10000D-01	5	3.24841D-07	1.10000D 00	0.0	0.0	0.0	1	9.52639D-03				
2	2.10000D-01	3	1.92284D-07	1.10000D 00	0.0	0.0	0.0	2	3.93887D-06	7	1.56341D 01	7	2.74539D 00
3	3.31000D-01	1	2.14828D-03	1.10000D 00	0.0	0.0	0.0						
3	3.31000D-01	5	6.41010D-07	1.10000D 00	0.0	0.0	0.0	1	5.34890D-04				
3	3.31000D-01	1	5.77034D-07	1.10000D 00	0.0	0.0	0.0	2	1.26332D-07	7	1.26554D 01	7	2.16295D 00
4	4.64100D-01	1	1.56703D-03	1.10000D 00	0.0	0.0	0.0						
4	4.64100D-01	5	2.01110D-07	1.10000D 00	0.0	0.0	0.0	1	4.25175D-04				
4	4.64100D-01	1	3.75154D-07	1.10000D 00	0.0	0.0	0.0	2	1.04412D-07	7	1.04926D 01	7	1.75533D 00
5	6.10510D-01	1	1.34768D-03	1.10000D 00	0.0	0.0	0.0						
5	6.10510D-01	5	5.23704D-07	1.10000D 00	0.0	0.0	0.0	1	3.74346D-04				
5	6.10510D-01	1	3.83334D-07	1.10000D 00	0.0	0.0	0.0	2	8.95652D-08	7	9.03227D 00	7	1.48496D 00
6	7.71561D-01	1	1.16634D-03	1.10000D 00	0.0	0.0	0.0						
6	7.71561D-01	5	6.71555D-07	1.10000D 00	0.0	0.0	0.0	1	3.35183D-04				
6	7.71561D-01	1	3.80247D-07	1.10000D 00	0.0	0.0	0.0	2	9.53810D-08	7	8.04727D 00	9	1.32357D 00
7	9.48717D-01	1	1.00824D-03	1.10000D 00	0.0	0.0	0.0						
7	9.48717D-01	5	4.26105D-07	1.10000D 00	0.0	0.0	0.0	1	3.05372D-04				
7	9.48717D-01	1	3.80979D-07	1.10000D 00	0.0	0.0	0.0	2	1.05167D-07	9	7.42715D 00	13	1.24194D 00
8	1.14359D 00	1	8.70563D-04	1.10000D 00	0.0	0.0	0.0						
8	1.14359D 00	5	6.04310D-07	1.10000D 00	0.0	0.0	0.0	1	2.74005D-04				
8	1.14359D 00	1	3.70629D-07	1.10000D 00	0.0	0.0	0.0	2	1.19575D-07	13	7.48618D 00	13	1.24002D 00
9	1.35795D 00	1	7.51655D-04	1.10000D 00	0.0	0.0	0.0						
9	1.35795D 00	6	1.24667D-07	1.10000D 00	0.0	0.0	0.0	1	2.51412D-04				
9	1.35795D 00	1	3.73684D-07	1.10000D 00	0.0	0.0	0.0	2	9.89227D-08	13	7.60064D 00	13	1.24356D 00
10	1.59374D 00	1	6.49755D-04	1.10000D 00	0.0	0.0	0.0						
10	1.59374D 00	6	2.19607D-07	1.10000D 00	0.0	0.0	0.0	1	2.35943D-04				
10	1.59374D 00	1	3.55348D-07	1.10000D 00	0.0	0.0	0.0	2	1.01473D-07	13	7.76437D 00	21	1.27138D 00

TABLE FOR SPECIAL MONITORING OF TEMPERATURES

NUMBER OF TIME STEPS	TIME	NODE NUMBERS AND TEMPERATURES							
10	1.59370 00	1 2.26562D 02	18 9.47883D 01	32 1.21935D 02	36 1.20418D 02	76 1.47573D 02			

11	1.85312D 00	1	5.63518D-04	1.10000D 00	0.0	0.0	0.0						
11	1.85312D 00	6	3.12309D-07	1.10000D 00	0.0	0.0	0.0	1	2.20234D-04				
11	1.85312D 00	1	3.39453D-07	1.10000D 00	0.0	0.0	0.0	2	1.04300D-07	21	8.14806D 00	21	1.32455D 00
12	2.13843D 00	1	5.33995D-04	1.10000D 00	0.0	0.0	0.0						
12	2.13843D 00	6	4.47636D-07	1.10000D 00	0.0	0.0	0.0	1	2.10690D-04				
12	2.13843D 00	1	3.61199D-07	1.10000D 00	0.0	0.0	0.0	2	1.13971D-07	29	8.66365D 00	29	1.40483D 00
13	2.45227D 00	1	5.17397D-04	1.10000D 00	0.0	0.0	0.0						
13	2.45227D 00	7	3.25699D-07	1.10000D 00	0.0	0.0	0.0	1	2.08749D-04				
13	2.45227D 00	1	3.73826D-07	1.10000D 00	0.0	0.0	0.0	2	1.08071D-07	69	9.43812D 00	69	1.51043D 00
14	2.79750D 00	1	4.96171D-04	1.10000D 00	0.0	0.0	0.0						
14	2.79750D 00	8	2.57087D-07	1.10000D 00	0.0	0.0	0.0	1	2.11682D-04				
14	2.79750D 00	1	3.89186D-07	1.10000D 00	0.0	0.0	0.0	2	1.09070D-07	69	1.04097D 01	69	1.64113D 00
15	3.17725D 00	1	4.70794D-04	1.10000D 00	0.0	0.0	0.0						
15	3.17725D 00	9	2.23566D-07	1.10000D 00	0.0	0.0	0.0	1	2.18338D-04				
15	3.17725D 00	1	4.10014D-07	1.10000D 00	0.0	0.0	0.0	2	1.10866D-07	69	1.14860D 01	69	1.78157D 00
16	3.59497D 00	1	4.41376D-04	1.10000D 00	0.0	0.0	0.0						
16	3.59497D 00	9	4.38604D-07	1.10000D 00	0.0	0.0	0.0	1	2.24113D-04				
16	3.59497D 00	2	2.21113D-07	1.10000D 00	0.0	0.0	0.0	2	1.52542D-07	69	1.26728D 01	69	1.93124D 00
17	4.05447D 00	1	4.09111D-04	1.10000D 00	0.0	0.0	0.0						
17	4.05447D 00	10	3.85137D-07	1.10000D 00	0.0	0.0	0.0	1	2.28731D-04				
17	4.05447D 00	2	2.26270D-07	1.10000D 00	0.0	0.0	0.0	2	1.59614D-07	69	1.39780D 01	69	2.08978D 00
18	4.55992D 00	1	3.91332D-04	1.19000D 00	0.0	0.0	0.0						
18	4.55992D 00	10	1.95579D-07	1.19000D 00	0.0	0.0	0.0	1	2.32390D-04				
18	4.55992D 00	2	2.71649D-07	1.19000D 00	0.0	0.0	0.0	2	1.59374D-07	69	1.54124D 01	69	2.25707D 00
19	5.11591D 00	1	4.70748D-04	1.19000D 00	0.0	0.0	0.0						

Table D-2 (Contd.)

```
19 5.11591D 00   10 4.00267D-07 1.19000D 00 0.0          0.0          0.0          1 2.35280D-04
19 5.11591D 00    2 3.66362D-07 1.19000D 00 0.0          0.0          0.0          2 1.93164D-07   69 1.69907D 01   69 2.43329D 00
20 5.72750D 00    1 5.64763D-04 1.19000D 00 0.0          0.0          0.0
20 5.72750D 00   11 3.28265D-07 1.19000D 00 0.0          0.0          0.0          1 2.38300D-04
20 5.72750D 00    2 5.33984D-07 1.19000D 00 0.0          0.0          0.0          2 2.21587D-07   69 1.87318D 01   69 2.61890D 00
  20      5.7275D 00     1 2.95458D 02    18 8.65445D 01   32 2.23019D 02   36 2.20696D 02    76 2.62632D 02

21 6.40025D 00    1 6.76147D-04 1.19000D 00 0.0          0.0          0.0
21 6.40025D 00   11 6.25727D-07 1.19000D 00 0.0          0.0          0.0          1 2.42327D-04
21 6.40025D 00    3 3.92551D-07 1.19000D 00 0.0          0.0          0.0          2 3.13356D-07   69 2.06586D 01   69 2.81458D 00
22 7.14027D 00    1 8.07741D-04 1.19000D 00 0.0          0.0          0.0
22 7.14027D 00   12 5.61071D-07 1.19000D 00 0.0          0.0          0.0          1 2.49105D-04
22 7.14027D 00    3 6.10101D-07 1.19000D 00 0.0          0.0          0.0          2 3.75284D-07   69 2.27980D 01   73 3.03392D 00
23 7.95430D 00    1 9.63923D-04 1.19000D 00 0.0          0.0          0.0
23 7.95430D 00   13 5.43348D-07 1.19000D 00 0.0          0.0          0.0          1 2.61024D-04
23 7.95430D 00    3 9.48579D-07 1.19000D 00 0.0          0.0          0.0          2 4.59938D-07   72 2.51923D 01   74 3.27454D 00
24 8.84973D 00    1 1.15052D-03 1.19000D 00 0.0          0.0          0.0
24 8.84973D 00   13 1.02188D-06 1.19000D 00 0.0          0.0          0.0          1 2.77788D-04
24 8.84973D 00    4 9.09321D-07 1.19000D 00 0.0          0.0          0.0          2 6.55495D-07   74 2.79773D 01   75 3.52582D 00
25 9.83471D 00    1 1.37486D-03 1.19000D 00 0.0          0.0          0.0
25 9.83471D 00   14 1.06215D-06 1.19000D 00 0.0          0.0          0.0          1 3.02930D-04
25 9.83471D 00    5 7.95072D-07 1.19000D 00 0.0          0.0          0.0          2 8.42657D-07   74 3.10953D 01   75 3.78893D 00
26 1.09182D 01    1 1.65041D-03 1.19000D 00 0.0          0.0          0.0
26 1.09182D 01   15 1.32498D-06 1.19000D 00 0.0          0.0          0.0          1 3.36006D-04
26 1.09182D 01    5 1.30258D-06 1.19000D 00 0.0          0.0          0.0          2 1.06411D-06   74 3.45434D 01   75 4.05724D 00
27 1.21100D 01    1 2.16783D-03 1.27100D 00 0.0          0.0          0.0
27 1.21100D 01   13 1.55856D-06 1.27100D 00 0.0          0.0          0.0          1 5.09818D-04
27 1.21100D 01    5 1.49189D-06 1.27100D 00 0.0          0.0          0.0          2 1.32093D-06   74 3.83506D 01   75 4.32858D 00
28 1.34210D 01    1 2.14181D-02 1.34390D 00 0.0          0.0          0.0
28 1.34210D 01   12 7.08819D-06 1.34390D 00 0.0          0.0          0.0          1 3.13792D-03
28 1.34210D 01    4 6.82862D-06 1.34390D 00 0.0          0.0          0.0          2 9.84424D-06   74 4.07128D 01   43 4.40445D 00
29 1.48631D 01    1 1.80026D-02 1.34390D 00 0.0          0.0          0.0
29 1.48631D 01   14 6.43874D-06 1.34390D 00 0.0          0.0          0.0          1 3.28674D-03
29 1.48631D 01    6 7.84038D-06 1.34390D 00 0.0          0.0          0.0          2 1.32561D-05
29 1.48631D 01    1 4.68024D-06 1.34390D 00 0.0          0.0          0.0          3 1.61884D-07   74 4.17489D 01   35 4.43131D 00
30 1.64494D 01    1 7.85148D-03 1.34390D 00 0.0          0.0          0.0
30 1.64494D 01   15 6.59625D-06 1.34390D 00 0.0          0.0          0.0          1 3.10978D-03
30 1.64494D 01    8 5.84831D-06 1.34390D 00 0.0          0.0          0.0          2 1.79364D-05
30 1.64494D 01    1 2.67489D-06 1.34390D 00 0.0          0.0          0.0          3 1.17123D-07   75 4.24598D 01   35 4.32453D 00
  30      1.6449D 01     1 4.00000D 02    18 1.10518D 02   32 4.73722D 02   36 5.17506D 02    76 5.81680D 02

31 1.81943D 01    1 6.17039D-03 1.34390D 00 0.0          0.0          0.0
31 1.81943D 01   17 4.87164D-06 1.34390D 00 0.0          0.0          0.0          1 3.46182D-03
31 1.81943D 01    9 5.66925D-06 1.34390D 00 0.0          0.0          0.0          2 2.28593D-05
31 1.81943D 01    1 4.54524D-06 1.34390D 00 0.0          0.0          0.0          3 1.50226D-07   75 4.24948D 01   35 4.13503D 00
32 2.01138D 01    1 8.24218D-02 1.34390D 00 0.0          0.0          0.0
32 2.01138D 01   18 6.77046D-06 1.34390D 00 0.0          0.0          0.0          1 7.88619D-03
32 2.01138D 01    9 6.12991D-06 1.34390D 00 0.0          0.0          0.0          2 4.62855D-05
32 2.01138D 01    1 9.62168D-06 1.34390D 00 0.0          0.0          0.0          3 2.47719D-07   75 4.16024D 01    7 4.07081D 00
33 2.22252D 01    1 4.88219D-02 1.34390D 00 0.0          0.0          0.0
33 2.22252D 01   19 8.20678D-06 1.34390D 00 0.0          0.0          0.0          1 4.79692D-03
33 2.22252D 01   10 6.06038D-06 1.34390D 00 0.0          0.0          0.0          2 3.74397D-05
33 2.22252D 01    2 5.57033D-06 1.34390D 00 0.0          0.0          0.0          3 4.70196D-07    7 5.86849D 01    7 6.21542D 00
```

Table D-2 (Contd.)

```
***** TABLE   1 MUST BE EVALUATED FOR  2.454766990 01
      THE VALUE OF THE FUNCTION WILL BE  3.000000000 00 FOR ALL ARGUMENTS GREATER THAN  2.400000000 01
  34 2.45477D 01     1 3.83995D-02 1.40951D 00 0.0          0.0          0.0
  34 2.45477D 01    17 7.23152D-06 1.40951D 00 0.0          0.0          0.0              1 3.73145D-03
  34 2.45477D 01    10 5.18760D-06 1.40951D 00 0.0          0.0          0.0              2 5.37274D-05
  34 2.45477D 01     2 6.90651D-06 1.40951D 00 0.0          0.0          0.0              3 7.42289D-07     7 5.39488D 01     7 5.37946D 00
  35 2.71024D 01     1 1.12409D-01 1.46856D 00 0.0          0.0          0.0
  35 2.71024D 01    15 9.44622D-06 1.46856D 00 0.0          0.0          0.0              1 9.01474D-03
  35 2.71024D 01    10 8.64491D-06 1.46856D 00 0.0          0.0          0.0              2 1.38231D-04
  35 2.71024D 01     4 8.22076D-06 1.46856D 00 0.0          0.0          0.0              3 1.93322D-06    45 4.00534D 01    31 3.51865D 00
  36 2.85512D 01     1 2.32964D-02 1.52170D 00 0.0          0.0          0.0
  36 2.85512D 01    15 7.54117D-06 1.52170D 00 0.0          0.0          0.0              1 3.04230D-03
  36 2.85512D 01     8 5.30051D-06 1.52170D 00 0.0          0.0          0.0              2 3.09714D-05
  36 2.85512D 01     1 6.45047D-06 1.52170D 00 0.0          0.0          0.0              3 2.93182D-07    69 2.01430D 01    69 1.60974D 00
  37 3.00000D 01     1 7.09172D-03 1.52170D 00 0.0          0.0          0.0
  37 3.00000D 01    14 7.02583D-06 1.52170D 00 0.0          0.0          0.0              1 1.66497D-03
  37 3.00000D 01     8 5.98647D-06 1.52170D 00 0.0          0.0          0.0              2 1.56820D-05
  37 3.00000D 01     1 3.68098D-06 1.52170D 00 0.0          0.0          0.0              3 2.13275D-07    69 1.76341D 01    69 1.38691D 00
```

Table D-2 (Contd.)

TRANSIENT TEMPERATURE DISTRIBUTION AFTER 37 TIME STEPS. TIME = 3.00000D 01

GROSS GRID			1		2	3	4	5	6	7
			I		I	I	I	I	I	I
FINE GRID			1	2	3	4	5	6	7	8
		DISTANCE	1.00	1.50	2.00	2.75	3.25	3.75	4.50	5.50
1	1	1.50	600.00	600.00	600.00	600.00	157.98	152.54	0.0	0.0
2	2	2.25	627.83	621.92	602.34	509.92	164.51	161.18	0.0	0.0
3	3	3.00	659.60	652.08	627.86	519.51	179.64	179.18	174.47	166.34
4	4	3.50	688.51	682.05	661.26	568.27	187.40	185.87	181.70	175.11
5	5	4.00	721.85	717.88	706.28	674.32	725.14	742.98	740.56	734.92
6	6	4.75	771.44	770.70	769.84	769.01	777.98	783.53	783.94	770.18
	7	5.25	796.71	796.51	796.29	796.90	800.09	802.16	800.36	786.92
	8	5.75	814.88	814.84	814.82	815.24	816.11	816.12	812.61	798.76
	9	6.25	825.78	825.77	825.76	825.83	825.68	824.67	820.14	805.87
7	10	6.75	829.44	829.40	829.38	829.30	828.87	827.54	822.68	808.25

TEMPERATURES ON NUMBERED BOUNDARIES

BOUNDARY NUMBER	TEMPERATURE
1	600.000000
2	0.0
3	68.000000
4	0.0
5	100.000000

THE CURRENT TIME STEP (DELTAT) = 1.448781580 00

ELAPSED CPU TIME IS 5.45 SECONDS

THE MAXIMUM TEMPERATURE IS - 8.29406D 02 (+-0.1)

MAX. TEMP. APPEARS AT NODES - 69 70 71

THE MINIMUM TEMPERATURE IS - 1.52544D 02 (+-0.1)

MIN. TEMP. APPEARS AT NODES - 6

Table D-2 (Contd.)

```
 NO      TIME        NO     MAX HEAT      BETA       L1 NORM OF      RHO          RHO       NO   L1 NORM OF NODE    MAX TEMP    NODE MAX PERCENT
TIME                ITER    RESIDUAL                 TEMP DIFF  (ITERATION)  (JACOBI)   ITER    TEMP DIFF          CHANGE          TEMP CHANGE

***** TABLE  2 MUST BE EVALUATED FOR  3.14051218D 01
      THE VALUE OF THE FUNCTION WILL BE  1.12500000D 00 FOR ALL ARGUMENTS GREATER THAN  3.00000000D 01
  38 3.28102D 01     1 9.82254D-03 1.52170D 00 0.0          0.0          0.0
  38 3.28102D 01    17 6.81089D-06 1.52170D 00 0.0          0.0          0.0          1 2.62919D-03
  38 3.28102D 01    12 3.91981D-06 1.52170D 00 0.0          0.0          0.0          2 5.73313D-05
  38 3.28102D 01     2 9.49345D-06 1.52170D 00 0.0          0.0          0.0          3 7.05657D-07   73 2.98767D 01   73 2.31862D 00
  39 3.59015D 01     1 1.09939D-02 1.52170D 00 0.0          0.0          0.0
  39 3.59015D 01    18 5.56444D-06 1.52170D 00 0.0          0.0          0.0          1 2.61459D-03
  39 3.59015D 01    12 6.67944D-06 1.52170D 00 0.0          0.0          0.0          2 7.34246D-05
  39 3.59015D 01     3 7.16633D-06 1.52170D 00 0.0          0.0          0.0          3 1.16072D-06   74 2.86446D 01   74 2.17492D 00
  40 3.93019D 01     1 1.37368D-02 1.52170D 00 0.0          0.0          0.0
  40 3.93019D 01    18 6.91398D-06 1.52170D 00 0.0          0.0          0.0          1 2.68926D-03
  40 3.93019D 01    13 9.59260D-06 1.52170D 00 0.0          0.0          0.0          2 8.08299D-05
  40 3.93019D 01     3 8.12380D-06 1.52170D 00 0.0          0.0          0.0          3 1.44621D-06   74 2.66938D 01   74 1.98366D 00

                                                        TABLE FOR SPECIAL MONITORING OF TEMPERATURES
NUMBER OF     TIME      ========================================= NODE NUMBERS AND TEMPERATURES ==========================================
TIME STEPS
    40      3.9302D 01     1 6.00000D 02    18 2.15392D 02    32 7.35367D 02    36 8.08855D 02    76 8.88391D 02

  41 4.30423D 01     1 1.43233D-02 1.52170D 00 0.0          0.0          0.0
  41 4.30423D 01    17 6.69889D-06 1.52170D 00 0.0          0.0          0.0          1 2.72097D-03
  41 4.30423D 01    14 9.14462D-06 1.52170D 00 0.0          0.0          0.0          2 9.75306D-05
  41 4.30423D 01     4 9.14904D-06 1.52170D 00 0.0          0.0          0.0          3 1.95822D-06   74 2.44017D 01   74 1.77806D 00
  42 4.71568D 01     1 1.45120D-02 1.52170D 00 0.0          0.0          0.0
  42 4.71568D 01    18 8.69512D-06 1.52170D 00 0.0          0.0          0.0          1 2.73177D-03
  42 4.71568D 01    15 5.51601D-06 1.52170D 00 0.0          0.0          0.0          2 1.05990D-04
  42 4.71568D 01     5 9.31550D-06 1.52170D 00 0.0          0.0          0.0          3 2.43957D-06   73 2.18294D 01   74 1.56218D 00
  43 5.16827D 01     1 1.43320D-02 1.52170D 00 0.0          0.0          0.0
  43 5.16827D 01    20 8.03275D-06 1.52170D 00 0.0          0.0          0.0          1 2.68511D-03
  43 5.16827D 01    15 7.30928D-06 1.52170D 00 0.0          0.0          0.0          2 1.16668D-04
  43 5.16827D 01     6 6.33251D-06 1.52170D 00 0.0          0.0          0.0          3 2.96511D-06   73 1.90708D 01   25 1.37049D 00
  44 5.58414D 01     1 1.23002D-02 1.56953D 00 0.0          0.0          0.0
  44 5.58414D 01    19 7.31211D-06 1.56953D 00 0.0          0.0          0.0          1 2.01241D-03
  44 5.58414D 01    15 7.18124D-06 1.56953D 00 0.0          0.0          0.0          2 8.30087D-05
  44 5.58414D 01     5 8.80689D-06 1.56953D 00 0.0          0.0          0.0          3 2.03303D-06   73 1.38288D 01   25 1.01228D 00
  45 6.00000D 01     1 9.23541D-03 1.56953D 00 0.0          0.0          0.0
  45 6.00000D 01    19 5.48275D-06 1.56953D 00 0.0          0.0          0.0          1 1.53724D-03
  45 6.00000D 01    15 5.60884D-06 1.56953D 00 0.0          0.0          0.0          2 6.48695D-05
  45 6.00000D 01     4 8.79009D-06 1.56953D 00 0.0          0.0          0.0          3 1.52864D-06   73 1.10084D 01   25 8.17880D-01
```

Table D-2 (Contd.)

TRANSIENT TEMPERATURE DISTRIBUTION AFTER 45 TIME STEPS. TIME = 6.00000D 01

GROSS GRID			1		2	3	4	5	6	7
			I		I	I	I	I	I	I
FINE GRID			1	2	3	4	5	6	7	8
		DISTANCE	1.00	1.50	2.00	2.75	3.25	3.75	4.50	5.50
1	1	1.50	600.00	600.00	600.00	600.00	219.73	211.49	0.0	0.0
2	2	2.25	661.76	655.57	635.00	537.37	232.07	227.04	0.0	0.0
3	3	3.00	728.64	720.73	695.05	578.15	258.37	258.17	250.16	235.97
4	4	3.50	780.61	774.08	752.88	654.38	270.66	268.72	261.79	250.22
5	5	4.00	835.74	832.30	822.64	800.73	872.28	897.56	905.50	884.70
6	6	4.75	912.75	912.85	914.80	921.42	936.69	946.20	947.29	925.77
	7	5.25	951.34	951.80	953.56	959.07	965.69	969.85	967.52	946.02
	8	5.75	978.88	979.31	980.65	984.23	986.96	987.84	982.77	960.39
	9	6.25	995.33	995.67	996.65	998.89	999.80	998.96	992.21	969.04
7	10	6.75	1000.79	1001.10	1001.94	1003.72	1004.09	1002.71	995.48	971.93

TEMPERATURES ON NUMBERED BOUNDARIES

BOUNDARY NUMBER	TEMPERATURE
1	600.000000
2	0.0
3	68.000000
4	0.0
5	100.000000

THE CURRENT TIME STEP (DELTAT) = 4.15862786D 00

ELAPSED CPU TIME IS 7.40 SECONDS

THE MAXIMUM TEMPERATURE IS - 1.00409D 03 (+-0.1)

MAX. TEMP. APPEARS AT NODES - 73

THE MINIMUM TEMPERATURE IS - 2.11493D 02 (+-0.1)

MIN. TEMP. APPEARS AT NODES - 6

THE TRANSIENT CALCULATIONS HAVE BEEN COMPLETED.

FINAL TIME IS 6.00000D 01

NUMBER OF TIME STEPS COMPLETED = 45

Table D-2 (Contd.)

```
BEGIN THE STEADY STATE CALCULATIONS
NUMBER OF                                                    EXTRAPOLATION
ITERATIONS      CONVERGENCE        NODE       TEMPERATURE       FACTOR

       5       6.26798D-03          44       9.47868D 02     -9.55058D 00
      10       3.27042D-03          32       8.25744D 02      6.16181D 00

                                                  TABLE FOR SPECIAL MONITORING OF TEMPERATURES
NUMBER OF      TIME      ========================================= NODE NUMBERS AND TEMPERATURES ==============
ITERATIONS
    10      6.0000D 01     1 6.00000D 02   18 2.75047D 02   32 8.25744D 02   36 9.17966D 02   76 1.01279D 03

      15       2.49038D-03          5       2.38222D 02      4.89342D 01
      20      -9.17783D-04         74       1.04848D 03      3.43357D 01
    20      6.0000D 01     1 6.00000D 02   18 2.84977D 02   32 8.36302D 02   36 9.23119D 02   76 1.01270D 03

      25      -7.83535D-04         25       2.96629D 02      1.62372D 01
      30      -5.00904D-04         17       2.80756D 02      5.13247D 00
    30      6.0000D 01     1 6.00000D 02   18 2.80511D 02   32 8.31058D 02   36 9.18769D 02   76 1.00921D 03

      35       1.53024D-04         36       9.18514D 02     -5.26005D-01
      40       1.44923D-04         33       9.06900D 02      2.07799D 01
    40      6.0000D 01     1 6.00000D 02   18 2.80609D 02   32 8.30983D 02   36 9.19170D 02   76 1.01036D 03

      45       1.13629D-04         37       9.47071D 02     -6.56472D 01
      50      -4.03554D-05         76       1.01057D 03     -4.42326D-01
    50      6.0000D 01     1 6.00000D 02   18 2.81126D 02   32 8.31765D 02   36 9.19639D 02   76 1.01057D 03

      55      -3.47337D-05         72       1.04606D 03     -1.27609D 01
      60      -2.69891D-05          5       2.37633D 02      1.73717D 01
                                    BETA REDUCED TO  1.800
    60      6.0000D 01     1 6.00000D 02   18 2.81006D 02   32 8.31589D 02   36 9.19382D 02   76 1.01037D 03

      65      -7.31851D-06          5       2.37586D 02      1.54156D 00
```

Table D-2 (Contd.)

STEADY STATE TEMPERATURE DISTRIBUTION AFTER 65 ITERATIONS. TIME = 6.00000D 01

GROSS GRID			1		2	3	4	5	6	7
FINE GRID			1	2	3	4	5	6	7	8
		DISTANCE	1.00	1.50	2.00	2.75	3.25	3.75	4.50	5.50
1	1	1.50	600.00	600.00	600.00	600.00	237.59	228.54	0.0	0.0
2	2	2.25	670.11	663.85	643.07	544.20	251.60	246.07	0.0	0.0
3	3	3.00	745.56	737.57	711.55	592.61	281.09	280.97	271.94	255.94
4	4	3.50	803.11	796.58	775.35	675.53	294.66	292.61	284.82	271.74
5	5	4.00	863.50	860.19	850.98	831.55	907.37	934.07	942.22	919.37
6	6	4.75	947.12	947.39	949.88	957.95	974.58	984.80	985.75	964.98
	7	5.25	988.96	989.55	991.68	998.12	1005.39	1009.85	1007.09	983.19
	8	5.75	1018.81	1019.32	1020.90	1025.05	1028.05	1028.96	1023.20	998.27
	9	6.25	1036.63	1037.03	1038.18	1040.76	1041.76	1040.79	1033.18	1007.35
7	10	6.75	1042.55	1042.91	1043.89	1045.94	1046.35	1044.78	1036.56	1010.39

TEMPERATURES ON NUMBERED BOUNDARIES

BOUNDARY NUMBER	TEMPERATURE
1	600.000000
2	0.0
3	68.000000
4	0.0
5	100.000000

ELAPSED CPU TIME IS 10.08 SECONDS

THE MAXIMUM TEMPERATURE IS - 1.04635D 03 (+-0.1)

MAX. TEMP. APPEARS AT NODES - 73

THE MINIMUM TEMPERATURE IS - 2.28538D 02 (+-0.1)

MIN. TEMP. APPEARS AT NODES - 6

THE STEADY STATE CALCULATIONS HAVE BEEN COMPLETED.

NUMBER OF ITERATIONS COMPLETED = 65

```
TEST PROBLEM #4 FOR HEATINGS
      100         7        11         3                2         1         5   CARD2
        7         7                   3         6                             CARD3
                  2                             10                            CARD4
       -2                                                                60.0   CARD5
        1         1       1.0       2.0       1.5      6.75                    R1
        1         1                           1                               R2
        2         1       2.0       5.5      4.75      6.75                    R1
        1         2                   5                                       R2
        3         1      3.25      3.75       1.5      2.25                    R1
        1                             3                                       R2
        4         1      3.25      3.75      2.25       3.0                    R1
        1                                                                     R2
        5         1      3.25       4.5       3.0       3.5                    R1
        1                                                                     R2
        6         1       4.5       5.5       3.0       3.5                    R1
        1                                       3                             R2
        7         2       2.0      2.75       1.5       4.0                    R1
        1                             3         1                             R2
        8         2       2.0      2.75       4.0      4.75                    R1
        1                                                                     R2
        9         2      2.75       5.5       4.0      4.75                    R1
        1                             5                                       R2
       10                2.75      3.25       1.5       3.5                    R1
        1                   2         2                                       R2
       11         3      3.25       5.5       3.5       4.0                    R1
        1                                       4         4                   R2
        1 IRON                      .2801      .116         3                   M
        2 STAINLSS                  .2824       .11        -4                   M
        3 AIR                                  0.25        -5        -6         M
        1       1.0         3                                                 G
        2       1.0         3                                                 G
        1                             2                                       I
        1         2     200.0        -1                                       B1
                                                                              B2
        2         3                                                           B1
          1.58D-13                                                            B2
        3         1      68.0                                                 B1
   6.0D-3                                                                     B2
        4         3                                                           B1
          1.58D-13 2.56D-05      0.33                                         B2
        5         1     100.0                                                 B1
          1.58D-13 2.56D-05      0.33                 1                       B2
                                                 1                            B3
      1.0       2.0       2.75      3.25      3.75      4.5       5.5         L1
        2         1          1         1         1         1                   N1
      1.5      2.25       3.0       3.5       4.0      4.75      6.75         L2
        1         1          1         1         1         4                   N2
        1         2                                                           A1
        4      0.03          5 .0087266                                       A2
```

Fig. D-4. Input Data for Test Problem Number 4 for HEATING5

```
         2         2                                                                  A1
         1     235.0         2     -20.0                                            A2
         3                                                                          A1
         1         4                                                              T1
       0.0       1.0      12.0       2.0      18.0       2.0      24.0       3.0   T2
         2         3                                                              T1
       0.0       1.0      12.0       1.5      30.0     1.125                      T2
         3         3                                                              T1
       0.0     .0296     752.0     .0264    1832.0     .0222                      T2
         4         3                                                              T1
       0.0      .013     752.0     .0153    1832.0      .025                      T2
         5         4                                                              T1
       0.0  1.82D-5     500.0   3.41D-5    1000.0   4.68D-5    1500.0   5.75D-5   T2
         6         4                                                              T1
       0.0   5.0D5      500.0   2.39D-5    1000.0   1.57D-5    1500.0   1.17D-5   T2
      30.0      60.0                                                              0
         5         1        18        32        36        76                      S
  1.0D-05   1.0D-3   1.0D-5                                                       IP
       0.1       1.1                                                            TP
```

Fig. D-4 (Contd.)

```
      SUBROUTINE HEATGN(RVALUE,R,TH,Z,TIM,TSN,VALUE,NUMBER,N)
C ***************************************************************************
C         THIS USER-SUPPLIED ROUTINE IS DESIGNED TO CALCULATE THE TWO
C         HEAT GENERATION RATES AS A SUM OF EXPONENTIALS FOR SAMPLE
C         PROBLEM NUMBER 4 IN THE HEATING5 USER'S MANUAL, ORNL/CSD/TM-15.
C ***************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL MESSAG
      DIMENSION NP(2),C(3,2),XLAMDA(3,2)
      DATA NP/3,2/ , C/0.5D0,0.3D0,0.2D0,0.6D0,0.4D0/
     1  , XLAMDA/1.15525D-2,2.31049D-2,4.62098D-2,1.15525D-2
     2  , 4.62098D-2/
      DATA MESSAG/.FALSE./
C        ON THE FIRST CALL TO THIS SUBROUTINE, MESSAG IS INITIALIZED TO
C        FALSE CAUSING THE FOLLOWING MESSAGE TO BE PRINTED OUT.  ON
C        SUBSEQUENT CALLS, THE WRITE STATEMENT IS BYPASSED.
      IF(MESSAG)GO TO 10
      MESSAG=.TRUE.
      WRITE(6,1000)
 1000 FORMAT('0***** IF THE INPUT DATA SPECIFIES THE FIRST TWO HEAT GENE
     1RATION RATES AS USER-SUPPLIED FUNCTIONS,'/
     2 ' ***** THIS ROUTINE WILL EVALUATE THEM AS SUMS OF EXPONENTIALS A
     3CCORDING TO THE FOLLOWING EXPRESSION.'/
     4 '0***** Q(TIM) = SUM(C(I)*EXP(-XLAMDA(I)*TIM))'/
     5 '0***** WHERE C(I) AND XLAMDA(I) ARE DEFINED AS'/
     6 '0***** FUNCTION NO.    I        C(I)         XLAMDA(I)')
      DO 5 J=1,2
      K=NP(J)
    5 WRITE(6,1010)J,(I,C(I,J),XLAMDA(I,J),I=1,K)
 1010 FORMAT('0',I12,I10,1P2D15.6/(' ',I22,2D15.6))
C
   10 IF(NUMBER.LT.1 .OR. NUMBER.GT.2)GO TO 900
      K=NP(NUMBER)
      SUM=0.0D0
      DO 20 I=1,K
   20 SUM=SUM+C(I,NUMBER)*DEXP(-XLAMDA(I,NUMBER)*TIM)
      RVALUE=SUM
      RETURN
  900 WRITE(6,9005)NUMBER
 9005 FORMAT('0***** THIS ROUTINE IS TRYING TO EVALUATE HEAT GENERATION
     1FUNCTION NUMBER',I10,' AS A USER-SUPPLIED FUNCTION.'/' ***** HOWEV
     2ER, THIS FUNCTION HAS NOT BEEN DEFINED HERE, SO THE CALCULATIONS W
     3ILL BE TERMINATED.')
      STOP
      END
```

Fig. D-5. User-Supplied Subroutine HEATGN to Calculate Heat Generation Rates for Test Problem Number 4 for HEATING5

```
      SUBROUTINE CONDTN(RVALUE,R,TH,Z,TIM,TSN,VALUE,NUMBER,N)
C ***************************************************************************
C THIS USER-SUPPLIED ROUTINE IS DESIGNED TO CALCULATE THE ANISOTROPIC
C THERMAL CONDUCTIVITY FOR A MATERIAL WITH THE CONDUCTIVITY ALONG THE
C Y-AXIS EQUAL TO TWICE THAT ALONG THE X-AXIS AS GIVEN IN TABULAR
C FUNCTION NUMBER #3.
C DESIGNED FOR SAMPLE PROBLEM NUMBER 4 IN THE HEATING5 USER'S MANUAL,
C ORNL/CSD/TM-15.
C ***************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /THBSBC/ NBDTP,NDIR
      LOGICAL MESSAG
      DATA MESSAG/.FALSE./
C
C        ON THE FIRST CALL TO THIS SUBROUTINE, MESSAG WILL BE FALSE
C        WHICH WILL CAUSE THE FOLLOWING MESSAGE TO BE PRINTED OUT.
C        ON SUSEQUENT CALLS, THE WRITE STATEMENT WILL BE BYPASSED.
      IF(MESSAG)GO TO 10
      MESSAG=.TRUE.
      WRITE(6,1000)NUMBER
 1000 FORMAT('0***** THE INPUT DATA SPECIFIES THE THERMAL CONDUCTIVITY O
     1F MATERIAL',I3,' TO BE A USER-SUPPLIED SUBROUTINE.'/
     2' ***** THIS ROUTINE WILL EVALUATE THE TEMPERATURE-DEPENDENT THERM
     3AL CONDUCTIVITY'/' ***** ALONG THE X-AXIS ACCORDING TO TABULAR FUN
     4CTION NUMBER 3.'/' ***** THE CONDUCTIVITY ALONG THE Y-AXIS WILL BE
     5 TWICE THAT ALONG THE X-AXIS.')
   10 IF(NUMBER.NE.1)GO TO 900
      CALL TABLE(-3,VALUE,TSN,RVALUE,N,NUMBER)
      IF(NDIR.EQ.2)RVALUE=2.0D0*RVALUE
      RETURN
  900 WRITE(6,9005)NUMBER
 9005 FORMAT('0***** CONDTN IS TRYING TO EVALUATE THE THERMAL CONDUCTIVI
     1TY FOR MATERIAL NUMBER',I9,' AS A USER-SUPPLIED SUBROUTINE.'/
     2' ***** HOWEVER, THIS PARAMETER HAS NOT BEEN DEFINED HERE, SO THE
     3CALCULATIONS WILL BE TERMINATED.')
      STOP
      END
```

Fig. D-6 User-Supplied Subroutine CONDTN to Calculate Anisotropic Thermal Conductivity for Iron for Test Problem Number 4 for HEATING5

APPENDIX E

GENERAL FLOW CHARTS FOR HEATING5

This appendix is designed to give a general idea of how the subroutines in HEATING5 are interconnected. The overall flow of the HEATING5 code is depicted in Fig. E-1. The call to subroutine INPUT causes most of the input data to be read and printed out in tabular form. Many variables and arrays are also initialized. The call to subroutine POINTS generates the nodal connections, initializes the temperatures and calculates the thermal conductances, capacitances and heat generation rates. The call to subroutine CALQLT then calculates the temperature distribution according to the method chosen and writes out the results. The routines called from subroutines THRMPR and CALQLT are depicted in greater detail in Figs. E-2 through E-6. Entry points are listed below names of subroutines. Table E-1 lists names of subroutines and entry points in alphabetical order along with a brief description of the purpose of each routine, a list of subroutines which references it, and a list of subroutines and entry points that are called by the routine.

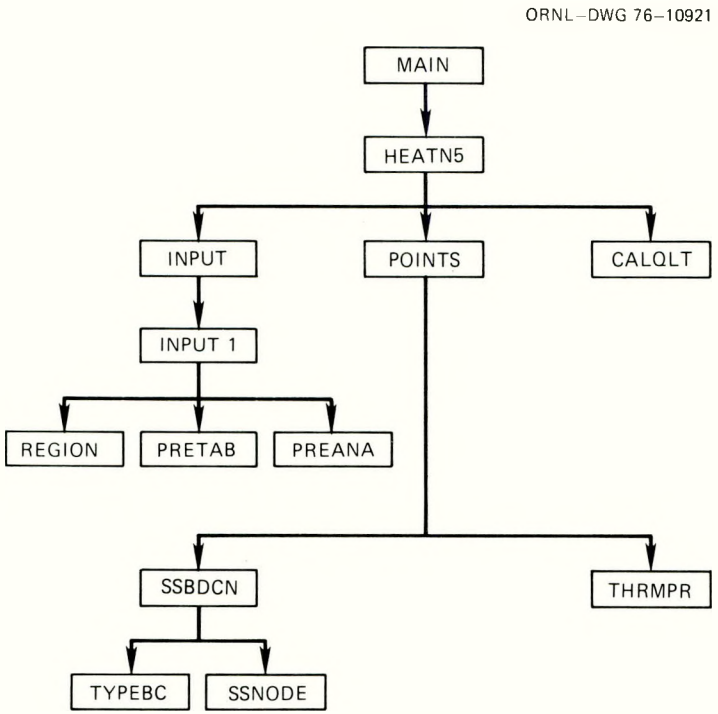

Fig. E-1. General Flow Chart of HEATING5

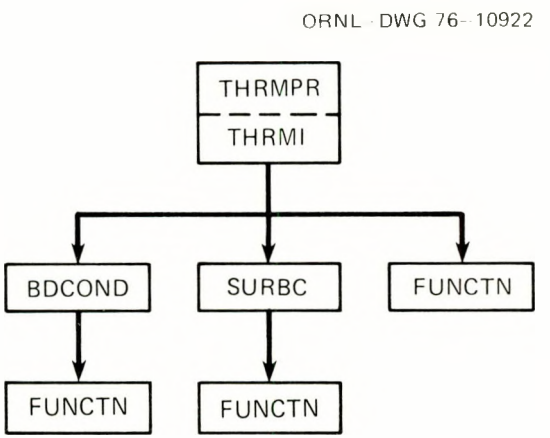

Fig. E-2. Chart Indicating Routines Called by Subroutine THRMPR

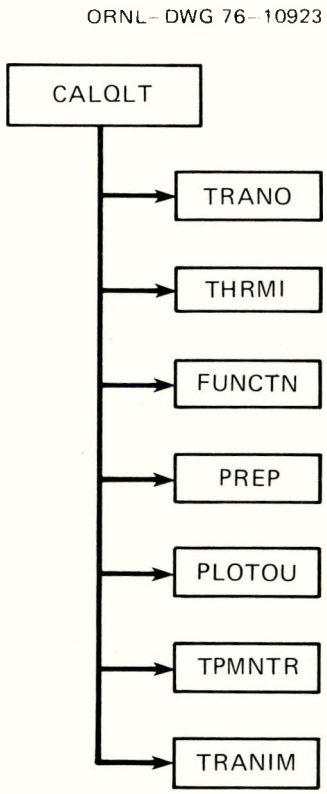

Fig. E-3. Chart Indicating Routines Called by Subroutine CALQLT

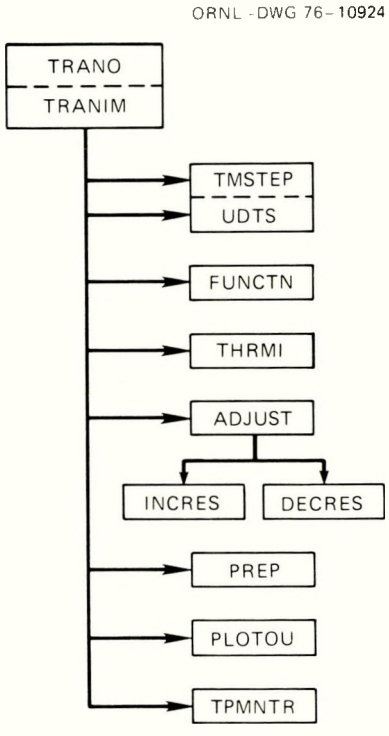

Fig. E-4. Chart Indicating Routines Called by Subroutine TRANO

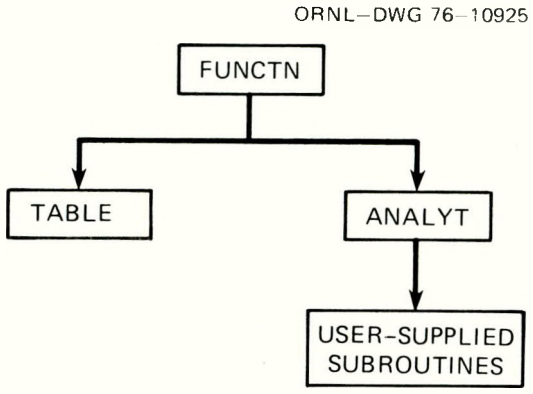

Fig. E-5. Chart Indicating Routines Called by Subroutine FUNCTN

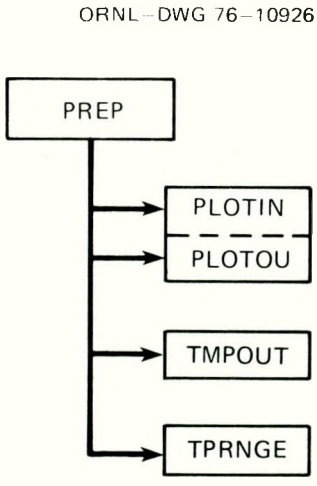

Fig. E-6. Chart Indicating Routines Called by Subroutine PREP

Table E-1. Summary of Subroutines and Entries in HEATING5

ADJUST – Adjusts the time step size for the implicit transient calculations based on any constraints on the maximum temperature change or maximum percentage of relative change in temperature at a node from one time level to the next. Requires that calculations continue at next time level with larger time step size or be repeated for this time level with smaller time step size. Called from TRANO. Calls INCRES and DECRES.

ANALYT – Entry in subroutine PREANA. Calculates value of a parameter according to built-in analytical function. May call user-supplied subroutine to evaluate a parameter. Called from FUNCTN. Calls user-supplied subroutines.

BDCOND – Calculates effective conductance from a surface node to a boundary node according to the associated boundary condition. Called from THRMPR. Calls FUNCTN.

CALQLT – Controls flow of calculational techniques after variables have been initialized. Contains steady-state and explicit transient algorithms. Calls routine for implicit transient algorithm where appropriate. Called from HEATN5. Calls FUNCTN, PLOTOU, PREP, THRM1, TPMNTR, TRANIM, and TRANO. Calls ORNL-dependent system routine ICLOCK.

DECRES – Attempts to decrease time step size for implicit transient calculations proportional to how much the temperature change or percentage of relative change in temperature exceeds the criterion. Called from ADJUST.

FUNCTN – Computes value of a parameter which can be time-, temperature-, and/or position-dependent. Value of parameter is product of functions of the independent variables–time, temperature, X or R, Y or θ, and Z. Called from BDCOND, CALQLT, SURBC, THRMPR, and TRANO. Calls ANALYT and TABLE.

HEATN5 – Locates position of each array which is variably dimensioned as a function of the maximum number of nodes. Controls overall flow of calculations. Called from MAIN. Calls CALQLT, INPUT, and POINTS.

INCRES – Attempts to increase time step size for implicit transient calculations proportional to how much the temperature change or percentage of relative change in temperature is less than the criterion. Called from ADJUST.

INPUT – First of two routines which reads the input data and prints it in tabular form. Generates fine grid lines. Called from HEATN5. Calls INPUT1. Calls ORNL-dependent systems routines ICLOCK, IDAY, MODEL, and TIME.

INPUT1 – Second of two routines which reads the input data and prints it in tabular form. (IT card images are read in subroutine THRMPR. Input data for implicit technique for transient problems are read in subroutines TRANO and TMSTEP.) Calculates factors (G arrays) which are a function of the grid spacing. Locates fine grid lines bounding each region. Called from INPUT. Calls PREANA, PRETAB, and REGION.

Table E-1 (Contd.)

MAIN	– Allocates core for arrays which are variably dimensioned as a function of maximum number of nodes. Calls HEATN5.
PLOTIN	– Creates data set containing identification as well as the temperature distribution as a function of time to be used by the code HEATPLOT to generate various plots. Called from PREP. Entry PLOTOU called from CALQLT, PREP, and TRANO.
PLOTOU	– Entry in subroutine PLOTIN. Writes temperature distribution on data set at current time. Called from CALQLT, PREP and TRANO.
POINTS	– Generates array identifying neighbors of each node. Model may have cutouts and indentations but axes must be orthogonal. Called from HEATN5. Calls SSBDCN and THRMPR.
PREANA	– Calculates value of a parameter according to built-in analytical function. May call user-supplied subroutines to evaluate a parameter. This call actually initializes routine. Entry ANALYT actually performs calculations. Called from INPUT1. Entry ANALYT called from FUNCTN. Calls user-supplied subroutines.
PREP	– Serves as an interface between routines which calculate temperature distribution and subroutine TMPOUT which prints out the temperature distribution in a map. Locates temperatures in a plane and passes them to TMPOUT. Called from CALQLT and TRANO. Calls PLOTIN, PLOTOU, and TMPOUT and TPRNGE. Calls ORNL-dependent systems routine ICLOCK.
PRETAB	– Computes value of a parameter by linear interpolation from a table of data. This call actually initializes routine. Entry TABLE actually computes value. Called from INPUT1. Entry TABLE called from FUNCTN.
REGION	– Locates fine grid line nearest a region boudary if a fine grid line does not lie on the region boudary. Called from INPUT1.
SSBDCN	– Locates region boundaries which require surface-to-surface heat transfer. Then, connects each node on one surface with the corresponding node on the opposing surface. Called from POINTS. Calls SSNODE and TYPEBC.
SURBC	– Calculates effective conductance from corresponding nodes on parallel surfaces according to the associated surface-to-surface boundary condition. Called from THRMPR. Calls FUNCTN.
TABLE	– Entry in subroutine PRETAB. Calculates value of a parameter by linear interpolation from a table of data. Called from FUNCTN.
THRMPR	– Calculates the initial temperature, the effective thermal conductance, the heat generation rate and the effective thermal capacitance at each node. Also reads explicitly specified initial temperatures and melting ratios. Called from POINTS. Entry THRM1 called from CALQLT and TRANO. Calls BDCOND, FUNCTN, and SURBC.
THRM1	– Entry in subroutine THRMPR. Updates the effective thermal conductance, the heat generation rate and the effective thermal capacitance at each node as a function of temperature. Called from CALQLT and TRANO. Calls BDCOND, FUNCTN, and SURBC.

Table E-1 (Contd.)

TMPOUT	– Prints nodal temperatures from a plane in the form of a map indicating material boundaries and grid lines. Called from PREP. Calls ORNL-dependent systems routines ICOMPA and INTBCD.
TMSTEP	– Initializes, reads, prints and calculates time step information for the implicit algorithm for transient problems. Called from TRANO. Entry UDTS called from TRANO.
TPMNTR	– Prints temperatures of selected nodes as a function of number of iterations or time steps. Used for special monitoring of the temperature development. Called from CALQLT and TRANO.
TPRNGE	– Determines maximum and minimum temperatures in the distribution and the nodes where these temperatures occur. Prints this information in tabular form. Called from PREP.
TRANIM	– Entry in subroutine TRANO. Computes the transient temperature distribution using an implicit algorithm. Called from CALQLT. Calls ADJUST, FUNCTN, PLOTOU, PREP, THRM1, TMSTEP, TPMNTR, and UDTS. Calls ORNL-dependent system routine ICLOCK.
TRANO	– Initializes and reads data for implicit transient calculations. Entry TRANIM computes the transient temperature distribution using a linear combination of the Crank-Nicolson and the backwards Euler procedures. Resulting system of equations is solved by SOR with the acceleration parameter optimized. Called from CALQLT. Entry TRANIM called from CALQLT. Calls ADJUST, FUNCTN, PLOTOU, PREP, THRM1, TMSTEP, TPMNTR, and UDTS. Calls ORNL-dependent system routine ICLOCK.
TYPEBC	– Determines if there are any surface-to-surface boundary conditions in a particular region along a particular axis. Called from SSBDCN.
UDTS	– Entry in subroutine TMSTEP. Reads, prints and calculates time step for the implicit algorithm for transient problems. Called from TRANO.

INTERNAL DISTRIBUTION

1. P. Angelini
2. S. J. Ball
3. W. E. Baucum
4. B. R. Becker
5. J. P. Belk
6. C. R. Bieber
7. A. A. Brooks
8. W. A. Burnett
9. J. P. Callahan
10. H. P. Carter
11. K. W. Childs
12. R. D. Cheverton
13. H. C. Claiborne
14. J. C. Cleveland
15. J. H. Coobs
16. W. K. Crowley
17. F. L. Culler
18. T. C. Domm
19. D. C. Elrod
20. G. G. Fee
21. C. A. Flanagan
22. T. B. Fowler
23. Uri Gat
24. G. E. Giles
25. T. G. Godfrey
26. N. Hanus
27. C. B. Haynie
28. R. A. Hedrick
29. R. W. Henderson
30. H. W. Hoffman
31. R. L. Huddleston
32. T. J. Huxford
33. S. H. Jury
34. R. A. Just
35. M. J. Kania
36-37. P. R. Kasten
38. T. S. Kress
39. C. G. Lawson
40. G. H. Llewellyn
41. A. L. Lotts
42. W. J. McAfee
43. S. R. McNeany
44. S. L. Milora
45. H. Postma
46. D. E. Reagan
47. M. W. Rosenthal
48. J. P. Sanders
49. W. K. Sartory
50. L. B. Shappert
51. K. K. Shipley
52. I. I. Siman-Tov
53. M. Siman-Tov
54. W. A. Simpson
55. Julius Smith
56. R. E. Stamps
57. W. C. T. Stoddart
58. A. J. Szady
59. K. R. Thoms
60. D. B. Trauger
61-85. W. D. Turner
86. D. R. Vondy
87. J. W. Wachter
88. G. E. Whitesides
89. W. R. Williams
90. H. T. Yeh
91. CSD Library - ORGDP Site
92-116. CSD Library - ORNL Site
117. Central Research Library
118. Document Reference Section - Y-12
119. Laboratory Records Dept.
120. Laboratory Records - RC
121. ORNL Patent Office

EXTERNAL DISTRIBUTION

122. Director, Division of Nuclear Research and Application, ERDA, Washington, DC 20555

123. Assistant Director, Reactor Programs, NRA, ERDA, Washington, DC 20555

124. Chief, Gas-Cooled Thermal Reactors Branch, NRA, ERDA, Washington, DC 20555

125. Chief, Fast Gas Reactor Branch, NRA, ERDA, Washington, DC 20555

126. Director, Reactor Division, ERDA, ORO, Oak Ridge, TN 37830

127. Director, Research and Technical Support Division, ERDA, ORO, Oak Ridge, TN 37830

128-176. Technical Information Center, ERDA, Oak Ridge, TN 37830 (21 copies ACRS)

177. Assistant Director for Reactor Safety, NRC/NRR/DSS, Washington, DC 20555

178, Assistant Director for Special Projects, NRC/NRR/DPM, Washington, DC 20555

179. Chief, Reactor System Branch, NRC/NRR/DSS, Washington, DC 20555

180. Chief, Analysis Branch, NRC/NRR/DSS, Washington, DC 20555

181. Chief, Core Performance Branch, NRC/NRR/DSS, Washington, DC 20555

182. Chief, Special Reactors Branch, NRC/NRR/DPM, Washington, DC 20555

183. Director, Office of Nuclear Regulatory Research, NRC, Washington, DC 20555

184. Assistant Director, Research Coordination, NRC/RES, Washington, DC 20555

185. Assistant Director, Advanced Reactor Safety Research, NRC/RES, Washington, DC 20555

186. Chief, Experimental Gas-Cooled Reactor Safety Research Branch, NRC/RES, Washington, DC 20555

187-189. Public Document Room USNRC, 1717 H St., Washington, DC 20555

190. R. E. Ireland, NRC/NRR/DSS, Washington, DC 20555

191. M. A. McCoy, NRC/NRR/DSS, Washington, DC 20555

192. L. E. Phillips, NRC/NRR/DSS, Washington, DC 20555

193. W. Kirk, Los Alamos Scientific Laboratory, P. O. Box 1663, Los Alamos, NM 87544

194. D. Schweitzer, Brookhaven National Laboratory, Upton, Long Island, NY 11973

195-200. General Atomic Company, Library, Room T-109, P. O. Box 81608, San Diego, CA 92138

201. J. Augerat, Laboratorie de Physique Nucléaire, 24 Av. des Landais B-P 45 63170 - Aubiere, FRANCE

202. B. D. Belt, Southern Services, Inc., Perimeter Center, East, P. O. Box 720071, Atlanta, GA 30346

EXTERNAL DISTRIBUTION (Contd.)

203. J. S. Crowell, International Atomic Energy Agency, Karntner Ring 11, P. O. Box 590, A-1011, Vienna, AUSTRIA

204. Ferruccio Gera, Laboratorio Rifiuti Radioattini, CNEN-CSN Casaccia, C.P. 2400, 00100 Rome, ITALY

205. D. Hodgkinson, Theoretical Physics Division, AERE Harwell, Oxfordshire, OX11 ORA

206. Juris Kaugerts, Brookhaven National Laboratory, Associated Universities, Inc., Upton, Long Island, NY 11973

207. W. H. Lake, Transportation Branch, Division of Materials and Fuel Cycle Facility Licensing, U. S. Nuclear Regulatory Commission, Washington, DC 20555

208. R. H. Odegaarden, Transportation Branch, Division of Materials and Fuel Cycle Facility Licensing, U. S. Nuclear Regulatory Commission, Washington, DC 20555

209. Clarence Oster, Battelle Northwest Laboratories, P. O. Box 999, Richland, WA 99352

210. L.D. Rickertsen, Science Applications, Inc., Oak Ridge, TN 37830

211. Virgil Sackschewsky, Rockwell International, Rocky Flats, CO 80401

212. C. C. Sartain, Dept. of Physics, Indiana State University, Terre Haute, IN 47809

213. T. S. Tully, Pullman-Swindell, 441 Smithfield Street, Pittsburgh, PA 15222

214. Wen-Shi Yu, Bldg. 129, Brookhaven National Laboratory, Upton, Long Island, NY 11973

☆ U.S. GOVERNMENT PRINTING OFFICE: 1977-748-189/207