A DIGITAL COMPUTER PROGRAM
TO DETERMINE THE TWO-DIMENSIONAL TEMPERATURE PROFILE IN GUN TUBES

TECHNICAL REPORT

Dr. William J. Leech
and
George E. Stiles

February 1972

RESEARCH DIRECTORATE

WEAPONS LABORATORY AT ROCK ISLAND

RESEARCH, DEVELOPMENT AND ENGINEERING DIRECTORATE

U. S. ARMY WEAPONS COMMAND

NATIONAL TECHNICAL INFORMATION SERVICE

Approved for public release, distribution unlimited.
DISPOSITION INSTRUCTIONS:

Destroy this report when it is no longer needed. Do not return it to the originator.

DISCLAIMER:

The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.
A DIGITAL COMPUTER PROGRAM TO DETERMINE THE TWO-DIMENSIONAL TEMPERATURE PROFILE IN GUN TUBES

This study was undertaken by the Research Directorate, Weapons Laboratory at Rock Island, to develop a digital computer program by which the two-dimensional temperature profile in gun tubes can be computed under realistic physical conditions. A mathematical model was presented in which variable geometry, temperature-dependent thermal properties, and variable conditions at the boundaries are considered. A numerical algorithm, in which the method of explicit finite-differences is used, was developed for the mathematical model and was programmed for the digital computer. A numerical example was computed to check the computer program. The program and all subroutines functioned properly. No numerical instability nor convergence problems were encountered. (U) (Leech, W. J. and Stiles, G. E.)
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
<tr>
<td>1. Heat Conduction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Temperature Distribution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Finite Differences</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Gun Tubes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A DIGITAL COMPUTER PROGRAM
TO DETERMINE THE TWO-DIMENSIONAL TEMPERATURE PROFILE
IN GUN TUBES

Dr. William J. Leech
and
George E. Stiles

February 1972
ABSTRACT

This study was undertaken by the Research Directorate, Weapons Laboratory at Rock Island, to develop a digital computer program by which the two-dimensional temperature profile in gun tubes can be computed under realistic physical conditions. A mathematical model was presented in which variable geometry, temperature-dependent thermal properties, and variable conditions at the boundaries are considered. A numerical algorithm, in which the method of explicit finite-differences is used, was developed for the mathematical model and was programmed for the digital computer. A numerical example was computed to check the computer program. The program and all subroutines functioned properly. No numerical instability nor convergence problems were encountered.
# CONTENTS

<table>
<thead>
<tr>
<th>Title Page</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>Contents</td>
<td>iii</td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Mathematical Model</td>
<td>2</td>
</tr>
<tr>
<td>Numerical Algorithm</td>
<td>6</td>
</tr>
<tr>
<td>Description of Computer Program</td>
<td>12</td>
</tr>
<tr>
<td>Numerical Example</td>
<td>14</td>
</tr>
<tr>
<td>Summary</td>
<td>17</td>
</tr>
<tr>
<td>Literature Cited</td>
<td>18</td>
</tr>
<tr>
<td>Appendix A - Digital Computer Program</td>
<td>19</td>
</tr>
<tr>
<td>Appendix B - Flow Chart of Main Program</td>
<td>32</td>
</tr>
<tr>
<td>List of Symbols Used in Text</td>
<td>34</td>
</tr>
<tr>
<td>Distribution</td>
<td>35</td>
</tr>
<tr>
<td>DD Form 1473 (Document Control Data - R&amp;D)</td>
<td>39</td>
</tr>
</tbody>
</table>
INTRODUCTION

Small caliber automatic weapons are subjected to extremely high operating pressures and temperatures. Energy from the hot propellant gas is absorbed by the gun tube at a much faster rate than it is dissipated to the surroundings. Temperature rises occur quite rapidly, and result in erosion or loss of strength of the gun tube material. High pressures may cause the tube to become ruptured when the temperatures are increased sufficiently. The important point is, therefore, that gun tube designers be able to predict the temperature distribution of a particular gun tube design. The purpose of the present work is to develop a digital computer program by which gun tube temperatures can be computed for physically realistic conditions. These physical conditions involve variable axial geometry, temperature-dependent thermal properties, variable firing schedules, and variable thermal boundary conditions at both the bore and exterior surfaces. Results obtained from the computer analyses may be used to determine areas of excessive temperature rise, to estimate maximum burst time, to provide information necessary for thermal stress analyses, and to indicate necessary changes for improved thermal performance.
MATHMATICAL MODEL

In this section, the physical mechanisms of heat transfer from the propellant gas to the gun tube are described and a mathematical formulation is given for the temperature distribution in the tube. An illustration of the gun tube is shown in Figure 1.

The gun tube material is considered to be isotropic, but the thermal properties, \( \rho(T) \), \( C(T) \), and \( K(T) \), are known functions of temperature. The assumption is that angular temperature variations are small, compared with radial and axial temperature variations. Thus, only a two-dimensional temperature field must be considered. Heat flows from the hot propellant gas, whose temperature is represented by \( T_g(r,z,t) \), to the tube, whose temperature is denoted by \( T(r,z,t) \). The assumption in this analysis is that the heat flux from the gas to the bore surface is specified or that the propellant gas temperature is a known function of time and position, and that a heat transfer coefficient, \( h(R_1,z,t,T) \), exists which is also a known function. And finally, the assumption is that continuous variations in the outside diameter of the tube may be adequately approximated by a finite number of step changes in the external diameter. The actual diameter as being approximated by three step changes is shown in Figure 1. The number of step changes may be greater or less than three, dependent upon the situation. Close approximation of any taper of the outside diameter by use of a greater number of step changes is possible. The analysis will be illustrated with the use of three step changes. However, the computer program was written so that any desired number of step changes in the external diameter could be handled.

The governing partial differential equation for the gun tube is given by

\[
\frac{\partial T}{\partial t} = \frac{K(T)}{\rho(T)C(T)} \left[ \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial z^2} \right] \\
+ \frac{1}{\rho(T)C(T)} \frac{\partial K}{\partial T} \left[ \left( \frac{\partial T}{\partial r} \right)^2 + \left( \frac{\partial T}{\partial z} \right)^2 \right]
\]

Equation 1 is nonlinear due to the presence of temperature-dependent properties. The boundary conditions for the tube illustrated in Figure 1 are

\[
r = R_1, \ 0 \leq Z \leq Z_3 \\
-K(T) \frac{\partial T}{\partial r} (R_1,Z,t) = q''(R_1,Z,t)
\]
Illustration of Gun Tube
\[ r = R_4, \quad 0 \leq Z \leq Z_1 \]
\[-K(T) \frac{\partial T}{\partial r}(R_4, Z, t) = h(T)[T(R_4, Z, t) - T_0] \]
\[+ \epsilon(T) \sigma[T^*(R_4, Z, t) - T_0^*] \quad (3)\]

\[ r = R_3, \quad Z_1 \leq Z \leq Z_2 \]
\[-K(T) \frac{\partial T}{\partial r}(R_3, Z, t) = h(T)[T(R_3, Z, t) - T_0] \]
\[+ \epsilon(T) \sigma[T^*(R_3, Z, t) - T_0^*] \quad (4)\]

\[ r = R_2, \quad Z_2 \leq Z \leq Z_3 \]
\[-K(T) \frac{\partial T}{\partial r}(R_2, Z, t) = h(T)[T(R_2, Z, t) - T_0] \]
\[+ \epsilon(T) \sigma[T^*(R_2, Z, t) - T_0^*] \quad (5)\]

\[ 0 < R < R_4, \quad Z = 0 \]
\[-K(T) \frac{\partial T}{\partial Z}(r, 0, t) = q^*(r, 0, t) \quad (6)\]

\[ R_3 \leq r \leq R_4, \quad Z = Z_1 \]
\[-K(T) \frac{\partial T}{\partial Z}(r, Z_1, t) = 0 \quad (7)\]

\[ R_2 \leq r \leq R_3, \quad Z = Z_2 \]
\[-K(T) \frac{\partial T}{\partial Z}(r, Z_2, t) = 0 \quad (8)\]

\[ R_1 \leq r \leq R_2, \quad Z = Z_3 \]
\[-K(T) \frac{\partial T}{\partial Z}(r, Z_3, t) = h(T)[T(r, Z_3, t) - T_0] \]
\[+ \epsilon(T) \sigma[T^*(r, Z_3, t) - T_0^*] \quad (9)\]
The initial conditions are

\[ T(r,Z,0) = T_l(r,Z) \]  \hspace{1cm} (10) \]

The heat flux given in Equation 2 may be a specified function, or may be expressed in terms of a heat transfer coefficient and the local difference between the bore surface temperature and the bore center line gas temperature. In the latter case, the boundary condition is given by

\[ q''(R_1,Z,t) = h(R_1,Z,t)[T_g(0,Z,t) - T(R_1,Z,t)] \]  \hspace{1cm} (11) \]

The heat flux given in Equation 6 must also be specified. This surface may lose energy to the surroundings, or exchange heat with some other section of the weapon. Boundary conditions for this surface must be specified on an individual basis.

The temperature-dependent thermal properties given in all the governing equations must be evaluated at the temperature of the point at which the equations are being evaluated. The heat fluxes at the boundary locations at which step changes have been used to approximate continuous variations in external diameter are assumed to be in the radial direction only. This is indicated in Equations 7 and 8. The radiation form factor for all other external surfaces has been taken as unity.

The set of equations given above cannot be solved analytically, so numerical techniques must be employed. The method of explicit finite differences was chosen to solve the equations. The details of the numerical algorithm are given in the following section.
NUMERICAL ALGORITHM

To determine the temperature distribution in the gun tube, the tube is first subdivided into a finite number of discrete lumps. The subdivision of the gun tube is illustrated in Figure 2. The tube has been divided into three sections in both the axial and the radial directions. The number of nodes in the first radial section is \( j_1 \), including the interface between sections 1 and 2. The second and third radial sections contain \( j_2 \) and \( j_3 \) nodes, respectively. In the first axial section, \( j_1 \) nodes are present, including the interface node. The second and the third axial sections contain \( j_2 \) and \( j_3 \) nodes, respectively. The spatial increments between the nodes are given by

\[
\Delta r_1 = \frac{(R_2 - R_1)}{(j_1 - 1)}
\]

(12)

\[
\Delta r_2 = \frac{(R_3 - R_2)}{j_2}
\]

(13)

\[
\Delta r_3 = \frac{(R_4 - R_3)}{j_3}
\]

(14)

\[
\Delta z_1 = \frac{Z_1}{(j_1 - 1)}
\]

(15)

\[
\Delta z_2 = \frac{(Z_2 - Z_1)}{j_2}
\]

(16)

\[
\Delta z_3 = \frac{(Z_3 - Z_2)}{j_3}
\]

(17)

The nodal point locations are

\[
r_n = R_1 + \frac{(n-1)}{(j_1 - 1)} (R_2 - R_1), \ 1 \leq n \leq j_1
\]

(18)

\[
r_n = R_2 + \frac{(n-1)}{j_2} (R_3 - R_2), \ i_1 < n \leq i_1 + i_2
\]

(19)

\[
r_n = R_3 + \frac{(n-1-i_2)}{j_3} (R_4 - R_3), \ i_1 + i_2 < n \leq i_1 + i_2 + i_3
\]

(20)
The temperature of each lump is assumed to be uniform and equal to the temperature of its center or nodal point. The spatial derivatives appearing in the governing equations are approximated by finite difference relations, which have been determined from the simultaneous solutions of truncated Taylor series expansions. Equation 1, written in discrete difference notation, is

\[
\frac{\partial T}{\partial t} = \nabla \cdot \left( K \nabla T \right)
\]

(24)

Subscript \( n \) denotes the radial node location and subscript \( m \) denotes the axial node locations. The subscripts of the property values denote that they are evaluated at the temperature of node \( r,m \).

\[
K_{n,m} = K(T_{n,m})
\]

(25)

The boundary conditions must also be written in difference form. For example, Equation 2 is written as

\[
-K_{1,m} \left( \frac{\partial T}{\partial r} \right)_{1,m} = q''_{1,m}
\]

(26)

The spatial derivatives in Equations 24 and 26 are approximated by finite difference relations. A set of explicit algebraic equations result for the time rate of temperature change at each node. These rates of change are multiplied by a finite time increment to find the temperature at one increment of time later. The procedure is repeated until the final time period of interest has been reached. The finite difference expressions used to approximate the spatial derivatives are shown in Tables I and II. The particular derivatives shown are for the
radial direction. The expressions for the axial direction are of the same form, where the spatial increment is $\Delta Z$, the subscript $n$ is fixed, and the subscript $m$ is a variable. The appropriate expression to use depends on the node of interest. For interior nodes, Number 2 in Table I and Number 3 in Table II are used. At the bore surface, Number 1 or Table II is used for the second derivative; the first derivative is determined from the boundary condition. The correct expressions for the external boundary nodes are those of the heat flux boundary conditions and Number 4 of Table II. The derivatives for the nodes adjacent to the boundaries are given by Numbers 2 and 3 of Table I and Number 2 of Table II. At the interface nodes, by which sections are separated in which the increments between nodes may become changed in size, the correct expressions are Numbers 4 and 5 of Table I and Table II, respectively.

The maximum time increment, by which numerical stability is ensured for the linear diffusion equation with this algorithm, is given by

$$\Delta t = \frac{\varepsilon C \Delta X^2}{K \omega},$$

where $\Delta X$ is the distance between nodal points and $\omega$ is

$$\omega = 4 + 4 \frac{h\Delta X}{K}.$$

The present set of equations are nonlinear and the maximum allowable time interval would be expected to be less than that given by Equation 27.

The sufficient condition for numerical stability and convergence is that both the first and the second laws of thermodynamics be satisfied. The satisfaction of the first law was verified by the performance of an energy balance on the tube after each time interval. No attempt was made to check the satisfaction of the second law. The belief was that the satisfaction of the first law provided an adequate check of numerical stability and convergence.

The explicit finite difference algorithm, described in this section, was programmed for the digital computer. The computer program is described in the following section, and a program listing is given in Appendix A.
**TABLE I**

FINITE DIFFERENCE EXPRESSIONS FOR FIRST DERIVATIVES

<table>
<thead>
<tr>
<th>No.</th>
<th>((\partial T/\partial r)_{n,m})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(\frac{1}{6\Delta r} [-2T_{n-1,m} - 3T_{n,m} + 6T_{n+1,m} - T_{n+2,m}])</td>
</tr>
<tr>
<td>2</td>
<td>(\frac{1}{12\Delta r} [T_{n-2,m} - 8T_{n-1,m} + 8T_{n+1,m} - T_{n+2,m}])</td>
</tr>
<tr>
<td>3</td>
<td>(\frac{1}{6\Delta r} [T_{n-2,m} - 6T_{n-1,m} + 3T_{n,m} + 2T_{n+1,m}])</td>
</tr>
<tr>
<td>4</td>
<td>(\frac{1}{\Delta r_2(1+\frac{\Delta r_2}{\Delta r_1})} \left[ T_{n+1,m} - T_n \left(1 - \frac{\Delta r_2^2}{\Delta r_1^2}\right) - \frac{\Delta r_2^2}{\Delta r_1^2} T_{n-1,m}\right])</td>
</tr>
</tbody>
</table>
### TABLE II

FINITE DIFFERENCE EXPRESSIONS FOR SECOND DERIVATIVES

<table>
<thead>
<tr>
<th>No.</th>
<th>((\partial^2 T/\partial r^2)_{n,m})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[ \frac{1}{18\Delta r^2} \left[ -85T_{n,m} + 108T_{n+1,m} - 27T_{n+2,m} + 4T_{n+3,m} \right] - \frac{11}{3\Delta r} \left( \frac{\partial r}{\partial r} \right)_{n,m} ]</td>
</tr>
<tr>
<td>2</td>
<td>[ \frac{1}{\Delta r^2} \left[ T_{n+1,m} - 2T_{n,m} + T_{n-1,m} \right] ]</td>
</tr>
<tr>
<td>3</td>
<td>[ \frac{1}{12\Delta r^2} \left[ -T_{n-2,m} + 16T_{n-1,m} - 30T_{n,m} + 16T_{n+1,m} - T_{n+2,m} \right] ]</td>
</tr>
<tr>
<td>4</td>
<td>[ \frac{1}{18\Delta r^2} \left[ -85T_{n,m} + 108T_{n-1,m} - 27T_{n-2,m} + 4T_{n-3,m} \right] + \frac{11}{3\Delta r} \left( \frac{\partial T}{\partial r} \right)_{n,m} ]</td>
</tr>
<tr>
<td>5</td>
<td>[ \frac{2}{\Delta r_1 \Delta r_2 + \Delta r_2^2} \left[ \left( \frac{\Delta r_2}{\Delta r_1} \right) T_{n-1,m} - \left( \frac{\Delta r_2}{\Delta r_1} + 1 \right) T_{n,m} + T_{n+1,m} \right] ]</td>
</tr>
</tbody>
</table>
DESCRIPTION OF COMPUTER PROGRAM

A digital computer program was written for the evaluation of the numerical algorithm described in the previous section. The computer program comprises a main program and eight subroutines. The main program contains the input, the output, the logic operations, and the computation operations for temperature changes. Detailed calculations are performed in the subroutines. The name and the purpose of each subroutine is given below:

1. CONV - Provides external convection coefficients and emissivities.
2. QZSUB - Contains operations to compute the axial heat fluxes at the external surfaces due to both radiation and convection.
3. QRSUB - Contains the operations necessary to compute the radial heat fluxes at the surfaces due to both radiation and convection.
4. AXIDER - Specifies the operations for the computation of the spatial derivatives in the axial direction.
5. RADER - Provides computations for the spatial derivatives in the radial direction.
6. DKDT - Gives derivative of thermal conductivity with respect to time.
7. XKKS - Gives the calculation of thermal conductivity as a function of temperature.
8. LINEAR - Gives specific heat as a function of temperature.

The input to the digital computer program consists of seven READ statements whose required input data are listed below:

1. M - number of radial segments
   N - number of axial segments
   SIGMA - radiation coefficient
   TS - ambient temperature
   TTIME - termination time
   NP - iteration number at which printout is desired
2. JJS(I) - number of radial nodes in each segment
3. LLS(I) - number of axial nodes in each segment
4. RS(J) - radial boundaries of each segment
5. ZS(I) - axial boundaries of each segment
6. KRA(I) - number of radial segments in each axial segment
7. X(I), Y(I) - temperature versus specific heat data

Thermal property data in the subroutines are for SAE 4130 steel. If a different barrel material is to be analyzed, the functional relationships in these subroutines must be changed. The emissivities and external convection coefficients in the present subroutines are constant. If these values are not constant for any case being investigated, the proper functional relations must be added to the subroutines.

The output from the program consists of the time and the temperature at each node for those iterations for which printout is desired. A complete listing of the digital computer program, along with typical input and output data, is given in Appendix A.
NUMERICAL EXAMPLE

A numerical example was computed to check the digital computer program. The sole purpose of computing the numerical example was to ensure that the program and the subroutines were functioning properly. No specific weapon was considered. With reference to Figure 1; the geometric dimensions used in the example were

\[ R_1 = 0.625 \text{ inch} \]  \hspace{1cm} (29)
\[ R_2 = 0.845 \text{ inch} \]  \hspace{1cm} (30)
\[ R_3 = 0.940 \text{ inch} \]  \hspace{1cm} (31)
\[ R_u = 1.088 \text{ inch} \]  \hspace{1cm} (32)
\[ Z_1 = 3.16 \text{ inch} \]  \hspace{1cm} (33)
\[ Z_2 = 12.5 \text{ inch} \]  \hspace{1cm} (34)
\[ Z_3 = 42.0 \text{ inch} \]  \hspace{1cm} (35)

Thermal property data for SAE 4130 steel were obtained from Figure 2.013, of Aerospace Structural Metals Handbook. The data were adjusted to the curves, which are given below.

\[ K = (28.3 - 0.0087T) \frac{\text{BTU}}{\text{hr ft}^2 \text{ F}}, \ T \leq 1420^\circ \text{F} \]  \hspace{1cm} (36)

and

\[ K = (10.39 + 0.00347T) \frac{\text{BTU}}{\text{hr ft}^2 \text{ F}}, \ T > 1420^\circ \text{F} \]  \hspace{1cm} (37)

The density variations for SAE 4130 steel are small, and the following mean value of density was used.

\[ \rho = 490 \text{ lb/ft}^3 \]  \hspace{1cm} (38)
Tabular data for specific heat were used in conjunction with a linear interpolation subroutine. The specific heat data from Figure 2.015 of Aerospace Structural Metals Handbook\textsuperscript{2} are given below:

\begin{tabular}{|c|c|}
\hline
T, °F & C, BTU/lb °F \\
\hline
0 & 0.108 \\
200 & 0.112 \\
400 & 0.125 \\
600 & 0.132 \\
800 & 0.150 \\
1000 & 0.160 \\
1200 & 0.185 \\
1600 & 0.180 \\
2000 & 0.180 \\
2200 & 0.150 \\
\hline
\end{tabular}

The temperature of the surroundings was constant and equal to 70°F. A constant convection coefficient of 5 BTU/hr ft °F and a constant emissivity of 0.5 were prescribed at the external boundaries. An effective mean propellant gas temperature of 2000°F and an effective mean heat transfer coefficient at the bore surface of 200 BTU/hr ft °F were used in the calculations.

The computer program was run, with the use of the data given above, for a continuous firing burst of 12.5 seconds. All portions of the main program and its subroutines functioned properly. No numerical instability nor convergence problems were encountered. The temperatures at all nodes were printed at time intervals of approximately one second. The computed bore surface temperatures, as functions of time and position, are shown graphically in Figure 3. The bore surface temperature rises more rapidly at locations in which the barrel wall is thinnest, as would be expected. The effects of axial temperature gradients are minor, except where an abrupt change occurs in the external diameter. This indicates that, for this specific example, a less complicated and less expensive one-dimensional numerical program could be used over most of the axial length without the introduction of any major errors. The two-dimensional program could still be used in regions where a step change exists in diameter.

A summary of this investigation is given in the following section.
SUMMARY

A mathematical model to determine the two-dimensional temperature profile in a gun tube, under realistic physical conditions, is presented. Variable geometry, temperature-dependent thermal properties, and variable conditions at the boundaries were considered in the mathematical model. A numerical algorithm was developed for the mathematical model by use of the method of explicit finite differences. The numerical algorithm was programmed for evaluation by the digital computer. A numerical example was computed to check the computer program. The program and all its subroutines functioned properly. No numerical instability nor convergence problems were encountered.
LITERATURE CITED


APPENDIX A

DIGITAL COMPUTER PROGRAM
A DIGITAL COMPUTER PROGRAM TO DETERMINE TWO-DIMENSIONAL TEMPERATURE PROFILE IN GUN TUBES BY THE METHOD OF EXPLICIT FINITE DIFFERENCES.

DIMENSION T(40,30), OR1(40,30), OR2(40,30), OR3(40,30), OR4(40,30), OR5(40,30), OR6(40,30), OR7(40,30), OR8(40,30), OR9(40,30), OR10(40,30), OR11(40,30), OR12(40,30), OR13(40,30), OR14(40,30), OR15(40,30), OR16(40,30), OR17(40,30), OR18(40,30), OR19(40,30), OR20(40,30), OR21(40,30), OR22(40,30), OR23(40,30), OR24(40,30), OR25(40,30), OR26(40,30), OR27(40,30), OR28(40,30), OR29(40,30), OR30(40,30), OR31(40,30), OR32(40,30), OR33(40,30), OR34(40,30), OR35(40,30), OR36(40,30), OR37(40,30), OR38(40,30), OR39(40,30), OR40(40,30)

1  T(10,30), OR1(10,30), OR2(10,30), OR3(10,30), OR4(10,30), OR5(10,30), OR6(10,30), OR7(10,30), OR8(10,30), OR9(10,30), OR10(10,30), OR11(10,30), OR12(10,30), OR13(10,30), OR14(10,30), OR15(10,30), OR16(10,30), OR17(10,30), OR18(10,30), OR19(10,30), OR20(10,30), OR21(10,30), OR22(10,30), OR23(10,30), OR24(10,30), OR25(10,30), OR26(10,30), OR27(10,30), OR28(10,30), OR29(10,30), OR30(10,30), OR31(10,30), OR32(10,30), OR33(10,30), OR34(10,30), OR35(10,30), OR36(10,30), OR37(10,30), OR38(10,30), OR39(10,30), OR40(10,30)

2  D1(150), D2(150), D3(150), D4(150), D5(150), D6(150), D7(150), D8(150), D9(150), D10(150), D11(150), D12(150), D13(150), D14(150), D15(150), D16(150), D17(150), D18(150), D19(150), D20(150), D21(150), D22(150), D23(150), D24(150), D25(150), D26(150), D27(150), D28(150), D29(150), D30(150), D31(150), D32(150), D33(150), D34(150), D35(150), D36(150), D37(150), D38(150), D39(150), D40(150)

3  JJ(10), LL(10), SS(10), ZS(10), JZ(10), IZ(10), IZS(10), IZT(10), JZT(10), IZT(10), KRA(10), KRA(10), KRA(10), KRA(10), KRA(10), KRA(10), KRA(10)

4  R1(20), R2(20)

COMMON /BRI/JSU(10), N, NUMT, HP, JNSUM, LL(3), M, SIGMA, TS
READ 1, N, M, SIGMA, TS, TIME, HP, NR
PRINT 201, N, M, SIGMA, TS, TIME, HP

201 FORMAT (N, 3S, 4H N., 15, 5X, ON N., 15, 5X, ON SIGMA., E15.6, 9BS, 15 TS., E12.5, 5X, ON TIME., E12.5, 5X, ON HP., 10)
C MP - WHICH ITERATION DESIRE DERIVATIVE PRINTS.
C NR - MODULUS NR INDICATES WHICH ITERATIONS TEMP PRINT OUTS ARE DESIRED.

READ 21, JJJS(I), J = 1, M)
PRINT 202, JJJS(I), I = 1, M)

202 FORMAT (JIJS(1), J = 1, M)
PRINT 203, (JIJS(I), J = 1, M)
PRINT 204, (JIJS(I), J = 1, M)

204 FORMAT H3/ (JE15.7))
C ALL SEGMENTS X SEGMENTS OUT RADIALY MUST HAVE SAME RADIAL DIMENSION.

READ 2, 1 ZS(1), I = 1, M)
PRINT 205, (ZS(I), I = 1, M)

205 FORMAT 2S/ (JE15.7))
C ALL SEGMENTS X SEGMENTS OUT AXIALLY MUST HAVE SAME AXIAL DIMENSION.

READ 21, (KRA(I), I = 1, M)
PRINT 206, (KRA(I), I = 1, M)

206 FORMAT 2S/ (JE15.7))
C PRINT 330, KRA(1), KRA(2), KRA(3)

330 FORMAT (4H KRA(1) =, 14, 10X, 4H KRA(2) =, 14, 10X, 4H KRA(3) =, 14)

C KNT = M + N + 1
C M MUST BE LESS THAN OR EQUAL TO N.
C MUST HAVE N NUMBER OF JSUM(1)'S.
C KRA(I) IS NUMBER OF RADIAL SECTIONS PER EACH AXIAL SECTION.
DO 3 K = 1, M
KRAD = KRA(1)
JSUM(1) = 0
DO 3 J = 1, KRAD
3 JSUM(K) = JJS(J) + JSUM(K)

PRINT 332, JSUM(1), JSUM(2), JSUM(3)
332 FORMATION JSUM(1) = .14, 10X, 10H JSUM(2) = .14, 10X, 10H JSUM(3) = .14

C ISUM = 0
DO 4 I = 1, N
4 ISUM = ISUM + LLS(I)

C ISUM REPRESENTS THE TOTAL NUMBER OF AXIAL NODES
C JJS= NUMBER OF RADIAL NODES IN THE RESPECTIVE SEGMENTS.
C LLS- NUMBER OF AXIAL NODES IN THE RESPECTIVE SEGMENTS.
C N = NUMBER OF AXIAL SEGMENTS.
C M = MAXIMUM NUMBER OF RADIAL SEGMENTS.
C T(I, J) = TEMPERATURES.
C DR(I) = RADIAL INCREMENT CHANGE (DR(1) < DR(2) < DR(3))
C DZ(I) = AXIAL INCREMENT CHANGE (START WITH DZ(1) = DZ(2) = DZ(3))
C Tlie AND ERROR FOR CORRECT VALUES FOR DZ(I)'
C RS(I) = BOUNDARY RADIUS FROM BORE TO OUTSIDE.
C ZS(I) = AXIAL BOUNDARIES.
C TS = AMBIENT TEMPERATURE.
C HGI(T) = WHERE IT IS TEMP SUBSCRIPT AND MG IS THE CONVECTION COEFFICIENT
C OF THE GAS COMPUTED AS A FUNCTION OF TEMP.
C XX = THERMAL CONDUCTIVITY AS A FUNCTION OF TEMP.
C CP(T) = SPECIFIC HEAT AS A FUNCTION OF TEMP.
C RHO(T) = DENSITY AS A FUNCTION OF TEMP.
C EMISI(T) = EMISSIVITY AS A FUNCTION OF TEMP.
C EHC(T) = CONVECTION COEFFICIENT AS A FUNCTION OF TEMP.
C EMR(T) = RADIATION AS A FUNCTION OF TEMP.
C TIME IS TIME OF TERMINATION

C THE FOLLOWING COMPUTES THE AXIAL AND RADIAL CHANGES (DELTAS)
C DO 6 J = 1, N
6 KJS(J) = JJS(J)
DO 7 I = 1, M
7 XLS(I) = LLS(I)
DZ(I) = ZS(I) / (XLS(I) - 1.0)
DO 8 I = 2, N
8 DZ(I) = (ZS(I) - ZS(I-1)) / XLS(I)

C NEXT COMPUTE THE RADII
C DR(I) = (RS(2) - RS(I)) / (KJS(I) - 1.0)
C R(I) = RS(I)
C JJJ = 2
C JJJ = 0
DO 9 J = 1, N
9 IF(JJ(J) EQ. 1) GO TO 9
C DR(JK) = (RS(JK+1) - RS(JK)) / KJS(JK)
C JJK = JJK + JJS(JK-1)
9 JJJ = JJJ + JJS(JK)
DO 10 J = JJK + JJJ
10 RII = RII - 1.0

C CONTINUE
C R(JJ) = RS(JJ)
C TIME = 0.0
C NNTNT = 1
C TSB = TS + 460.0
C NNT = 0
J11(I) = JJS(I) + 1
MN = N - 1
DO 14 J = 1, MN
14 J11(J) = J11(J-1) + JJS(J)
IK = N - LLS(I)
IK = 1
NN = LLS(I)
DO 18 J = 1, MN
JSU = JSUM(I)
DO 17 I = IK, NN
DO 17 J = 1, JSU
17 T(I,J) = TS
IF(TL = EQ, N) GO TO 10
IK = IK + LLS(I)
NN = NN + LLS(I+1)
10 CONTINUE
60 CONTINUE
C NEXT COMPUTE THE MAXIMUM TIME INTERVAL.
C DX = DAI(I) (SINCE DAI(I) < DAI(I) OR DZ(I))
C DX = DAI(I)
C NOW LOCATE MAXIMUM XX(IT), AND MINIMUM XH(IT)
C XMAX = XX(IT)MAX
C XKMIN = XXH(IT)MIN
C XMAX = 26.8
C XKMIN = 12.0
C XMAX = 24.0
C XKMIN = 20.7
C NOW LOCATE RHON(T) MINIMUM, CPT(T) MINIMUM.
C RHOMIN = RHON(T)MIN
C CPTMIN = CPT(T)MIN
C RHOMIN = .400.0
C CPTMIN = .100
C DT(1) = RHOMIN * CPTMIN / XMR / XMAX * DX * DX
C XMR = 4. * 4. * XKMIN / XMAX * (AS(I) - AS(I)) / (XLS(I) - 1.0)
C DT(2) = RHOMIN * CPTMIN / XMR / XMAX * (AS(I) - AS(I)) / (XLS(I) - 1.0)
1 DO 19 J = 2, N
C XMR = 4. * 4. * XKMIN / XMAX * (AS(J+1) - AS(J)) / XLS(J)
19 DT(J+1) = RHOMIN * CPTMIN / XMR / XMAX * (AS(J+1) - AS(J)) * 2 / XLS(J) # 2
C XMR = 4. * 4. * XKMIN / XMAX * (SZ(I) - 6Z(IJ) / (XJS(I) - 1.0)
C DT(J+1) = RHOMIN * CPTMIN / XMR / XMAX * (SZ(I) - 6Z(IJ) / XJS(I) # 2
DO 190 J = 2, N
C XMR = 4. * 4. * XKMIN / XMAX * (SZ(I) - 6Z(IJ) / XJS(I)
190 DT(J+1) = RHOMIN * CPTMIN / XMR / XMAX * (SZ(I) - 6Z(IJ) / XJS(I) # 2
C /= XJS(I) # 2
C NOW MUST DETERMINE SMALLEST DT AND SAE VALUE WILL BE THE TIME
C INCREMENT.
C IOT = 0
20 IOT = IOT + 1
IPT = IOT * 1
IF(IPT = EQ, I) GO TO 50
DT(IPT) = DT(IPT)
50 KINT = KINT - 1
IF(IPT = NE, KINT) GO TO 20
C KINT MUST BE READ IN AS (N + R + 1)
DIM = DT(KINT1)
DIM = DIM / 1.5
TIME = TIME + DIM * 3600.0
C  NOW COMPUTE ALL VALUES DEPENDENT ON TEMPERATURE
C  SUBSCRIPT IT DESIGNATES WHICH TIME INTERVAL
C  XMC(I,J) MUST EITHER BE A TABLE READ IN OR A FUNCTION COMPUTATION.
C  NEXT COMPUTE THE HEAT FLUXES.
LL1 = LLS(I)
CALL QRUSUB1(OR,TSAB,JM,LLS)
C  CALL QZUSUBITSAB,JJS,LLS,T,QKRA)
C  NOW CHECK RADIAL TEMP AGAINST AXIAL TEMP DIFF AND SHOULD BE CLOSE.
C  ONLY NEED TO CHECK BORE TEMPS AXIALLY AGAINST ADJACENT RADIAL TEMPS.
CALL RADERIT(OR,DT,DTZ,OR,JJS,LLS)
C  NEXT COMPUTE AXIAL DERIVATIVES.
CALL AXDDERIT.D2.0U,DT,DT2,OR,JJS,LLS)
C  COMPUTE TEMP CHANGE AT EACH NODE
IF(KINT .NE. 1) GO TO 490
IF(MOD(KINT,NR) .NE. 0) GO TO 510
C
490 PRINT 500, TIME, NKONT
900 FORMAT(I5, 6H TIME = , 12E4, 10X, 18H ITERATION NUMBER = , I5/)
510 IFF = 0
DO 503 IK = 1, N
II = IFF + 1
IFF = IFF + LLS(1K)
JSU = JSUM(1K)
DO 434 J = 1, JSU
DO 434 I = IFF, JSU
IT = T(I,J)
CALL OKDITT, XXX, OKT, OKT, XRHO, XCP, X, Y)
C  OKDIT IS THERMAL CONDUCTIVITY, DENSITY, SPECIFIC HEAT SUBROUTIN
IF(NKONT .NE. 0) GO TO 803
PRINT 802, XXX, XRHO, XCP, DTR2(I,J), R(I,J), DTR2(I,J), DT2(I,J)
1 OKT, DTZ(I,J)
802 FORMAT(6H XXX = , 1E15.7, 5X, 7H XRHO = , 1E15.7, 5X, 6H XCP = , 1E15.7,
1 5X, 12H DTR2(I,J) = , 1E15.7 / 7H R(I,J) = , 1E15.7, 5X, 11H DTR2(I,J)
2=, 1E15.7, 5X, 12H DTZ(I,J) =, 1E15.7, 5X, 6H OKT =, 1E15.7 / 5X, 12H
3 DTZ(I,J) =, 1E15.7)
803 DDT COMPUTE TEMP CHANGE AT EACH NODE
IF(NKONT .NE. 0) GO TO 441
IF(MOD(NKONT,NR) .NE. 0) GO TO 503
C
441 CONTINUE
DO 501 I = IFF, IFF
PRINT 502, I
502 FORMAT(5X, 18H AXIAL LOCATION = , 15I)
PRINT 505, (TI(I,J), J = 1, JSU)
501 CONTINUE
503 CONTINUE
505 FORMAT(I8),15(I12.1)
560 NKONT = NKONT + 1
C  IF(KINT .LE. 500) GO TO 40
IF(TIME .LT. TTIME) GO TO 60
760 CALL EXIT
END
SUBROUTINE CONVITT, XXHC, EMISS, XK
EMISS = .5
XXHC = 5.0
CALL XKKSIT, XK, OKT)
RETURN
END

SUBROUTINE QZSUB(TSAB, JSU, LLS, T, QZ, KRA)
COMMON /BLKI/JSUNIBIT, N, NKONT, NP, ISUM, LLS, M, SIGMA, TS
DIMENSION OZMO(JO), TI40.
JSISI
JSISI.
KRAIB.
XI20I.

DO 90 IK = 1, N
ISU = 0
DO 5 IK = 1, KR
5 ISU = ISU + LLS(IK)
IF(KR .NE. N) GO TO 8
JSU = JSUM(N)
DO 12 J = 1, JSU
TABS = TISUM(J) * 460.0
T = TISUM(J)
CALL CONVITT(TT, XXHC, EMISS, XK)
XXHR = EMISS * SIGMA * I TABS ** 3 + TABS ** 2 * (TSAB) *
1 TABS * (TSAB) ** 2 + (TSAB) ** 3)
XXH = XXHC + XXHR
IF(NKONT .NE. 0) GO TO 9
PRINT 201, XK, XXHR, XXH
201 FORMAT(5H, XK =, E15.7, 5X, 7H XXHR =, E15.7, 5X, 6H XXH =, E15.7)
6 QZ(ISUM, J) = XXH / XK * (TISUM(J) - TS)
IF(NKONT .NE. 0) GO TO 12
PRINT 10, ISUM, J, QZ(ISUM, J)
10 FORMAT(3H, ISUM =, 15, 5X, 3H J =, 15, 10X, 11H QZ(ISUM, J) =, E12.4)
12 CONTINUE
GO TO 90
8 JSU = JSUM(KR)
JSUI = JSUM(KR+1)
IF(JSU .EQ. JSUI) GO TO 90
IF(JSU .LT. JSUI) GO TO 80
JSUI = JSU + 1
DO 70 J = JSU1, JSU
TABS = TISU(J) * 460.0
T = TISU(J)
CALL CONVITT(TT, XXHC, EMISS, XK)
XXHR = EMISS * SIGMA * I TABS ** 3 + TABS ** 2 * (TSAB) *
1 TABS * (TSAB) ** 2 + (TSAB) ** 3)
XXH = XXHC + XXHR
IF(NKONT .NE. 0) GO TO 9
PRINT 201, XK, XXHR, XXH
9 QZ(ISU, J) = XXH / XK * (TISU(J) - TS)
IF(NKONT .NE. 0) GO TO 70
PRINT 203, ISU, J, QZ(ISU, J)
203 FORMAT(5H, ISU =, 110, 5X, 4H J =, 110, 5X, 12H QZ(ISU, J) =, E15.7)
70 CONTINUE
GO TO 90
80 JSUM1 = JSU + 1
JSU1 = JSU + 1
DO 85 J = JSUM1, JSU1
TABS = TISUM1(J) * 460.0
T = TISUM1(J)
90 CONTINUE
CALL CONVITT, XXHC, EMISS, XXH
XXHR = EMISS * SIGMA  * (TABS ** 3 * TABS ** 2 * (TSAB) +
1   TABS * (TSAB) ** 2 * (TSAB) ** 3)
XXH = XXHC + XXHR
IF(NKONT .NE. 0) GO TO 85
PRINT 201, XXH
QZ(ISUL,J) = XXH / XXH * (TS - T(ISUL,J))
85 CONTINUE
90 CONTINUE
RETURN
END

SUBROUTINE ORSJBIT, OR.TSAB, JH.LLSI
COMMON /BLK1/JSUM(N), N, NKONT, NP, ISUM, LLI, M, SIGMA, TS
DIMENSION T(40,30), QZ(40,30), LLS(I), X(20), Y(20)
DO 20 I=1,ISUM
TABS = T(I,1) + 460.0
TT = T(I,1) + 460.0
CALL CONVITT, XXHC, EMISS, XXH
XXHR = EMISS * SIGMA  * (TABS ** 3 * TABS ** 2 * (TSAB) +
1   TABS * (TSAB) ** 2 * (TSAB) ** 3)
XXH = XXHC + XXHR
20 QR(I,J) = XXH / XXH * (T(I,J) - TS)
LI = LLI
IS = 0
DO 200 IK = 1,N
JSU = JSUM(IK)
T = JSU
IS = IS + 1
IS = IS + LLS(IK)
DO 200 I = 1, IS
TABS = T(I,JSU) + 460.0
TT = T(I,JSU)
CALL CONVITT, XXHC, EMISS, XXH
XXHR = EMISS * SIGMA  * (TABS ** 3 * TABS ** 2 * (TSAB) +
1   TABS * (TSAB) ** 2 * (TSAB) ** 3)
XXH = XXHC + XXHR
200 CONTINUE
RETURN
END

SUBROUTINE AXIDERT,Z2.D2Z,DTZ2.QZ,LLS
DIMENSION T(40,30), DZ(40,30), DTZ(40,30), DZ2(40,30), QZ(40,30),
LLS(I), X(20), Y(20)
COMMON /BLK1/JSUM(N), N, NKONT, NP, ISUM, LLI, M, SIGMA, TS
IS = 0
DO 320 IK = 1,N
JSU = JSUM(IK)
IF(IK .EQ. N) GO TO 150
C IF(JSU .LE. JSUM(IK+1)) GO TO 150
I = IS + 1
IS = IS + LLS(IK)
150 DO 320 J=1,JSU
IF(J .NE. I) GO TO 185
QZ(I,J) = 0.0
DZ2(I,J) = -QZ(I,J)
DZ2(I,J) = 1.0/18.0/DZ(1) ** 2 *(-85. * T(I,J) + 108.0 * T(2,J) -
25
1 27.0 * T(3,J) + 6.0 * T(4,J) + 11. / 3.0 * QZ(1,J) / DZ(1,J)
IF(MODINKONT.NP) .NE. 0) GO TO 152
PRINT 20, J, DTZ1(J), DTZ2(J)
20 FORMAT(3H DTZ1(1,J), 3H) = E15.6, 10X, 7H DTZ2 = E15.6)
152 J = 11 1
DTZ1(J) = 1./6. / DZ1(K) * (T(12-,1,J) * T(12,J) + 6. * T(12,J) - T(12,J))
DTZ2(J) = 1. / DZ1(K) * 2 * (T(12+1,J) - T(12,J) * T(12-1)
J)
IF(MODINKONT.NP) .NE. 0) GO TO 31
PRINT 22, J, DTZ1(J), DTZ2(J)
22 FORMAT (15H DTZ1(13,J) = E15.6, 10X, 7H DTZ2 = E15.6)
31 INT = 11 + 2
I$ = 15 - 2
DO 153 I = INT, 152
DTZ1(J) = 1. / 12. / DZ1(K) * T(11 - 2, J) - B. * T(11 - 1, J) + B.
10 T(11 + 1, J) = T(11 + 2, J)
C 153 DTZ2(J) = 1. / 12. / DZ1(K) * 2 * (T(11-2,J) + 10. * T(11-1)
J) - DTZ1(J) * 2 * (T(11+1,J) - 10. * T(11,J))
IF(MODINKONT.NP) .NE. 0) GO TO 153
PRINT 20, J, DTZ1(J), DTZ2(J)
153 CONTINUE
111 = INT + 1
DTZ1(J) = 1. / 6. / DZ1(K) * T(111-1,J) + 3. * T(111+1, J) + 2
1. * T(111,J) = T(111 + 2, J)
DTZ2(J) = 1. / DZ1(K) * 2 * (T(111+1,J) - 2. * T(111,J) + T(111-1
J))
IF(MODINKONT.NP) .NE. 0) GO TO 32
PRINT 22, J, DTZ1(J), DTZ2(J)
32 IF(IK <= N) GO TO 165
IF(J, GT, JSUM(1+1)) GO TO 165
DTZ1(1S,J) = 10. / DZ1(K) * 1. / DZ1(K) * (T(1S,J) - T(1S,J)
1. * DZ1(K) * T(1S,J) - DZ1(K) == 2 * / DZ1(K) * 2
I = DZ1(K) * 2 * (T(1S,J) - T(1S,J))
DTZ2(1S,J) = 2. * DZ1(K) * DZ1(K) + DZ1(K) * 2 * (DZ1(K) * DZ1(K)
1. * DZ1(K) + DZ1(K) + 2 * (T(1S,J) - T(1S,J))
IF(MODINKONT.NP) .NE. 0) GO TO 32
PRINT 22, J, DTZ1(J), DTZ2(J)
GO TO 320
160 IF(IK .EQ. 1) GO TO 320
IF(J, GE, JSUM(1-1)) GO TO 165
165 DTZ1(1S,J) = DZ1(1S,J)
DTZ1(1S,J) = 1. / 16. / DZ1(K) * 2 * (T(1S,J) + 108. * T(1S,J)
IF(MODINKONT.NP) .NE. 0) GO TO 320
PRINT 22, J, DTZ1(J), DTZ2(J)
GO TO 320
185 IF(J, GT, JSUM(-1)) GO TO 200
190 DTZ1(1S,J) = 1. / 6. / DZ1(K) * (T(1S,J) - T(1S-1,J) - 3. * T(1S,J)
IF(MODINKONT.NP) .NE. 0) GO TO 33
PRINT 22, J, DTZ1(J), DTZ2(J)
GO TO 152
200 IF(J, LE, JSUM(-1)) GO TO 190
DTZ1(1S,J) = DZ1(1S,J)
DTZ1(1S,J) = 1. / 18. / DZ1(K) * 2 * (T(1S,J) + 108. * T(1S,J)

IF MOD(NKONT, NP) .NE. 0 GO TO 34
PRINT 22, I, J, DTZ(I, J), DTZ2(I, J)
34 IF (I .NE. N) GO TO 150
GO TO 152
320 CONTINUE
RETURN
END

SUBROUTINE RADERIT, DR, DTR, DTR2, QR, JJ5, LLSI
DIMENSION T(40), DR(5), DTR(40, 30), DTR2(40, 30), QR(40, 30),
JJ5(18), LLSI(8), X(20), Y(20)
COMMON /BLK1/JSUNiait, NKONT, NP, ISUM, LL1, M, SIGMA, TS
C
XH = 200.0
TG = 2000.0
C
IFT = 1
ILA = LLSI(1)
DU 360 IK = 1, M
JSX = JSUM(1K)
DO 305 JR = 1, M
JSX = JSX - JJ5(JR)
IF (JSX .EQ. 0) GO TO 306
305 CONTINUE
JR = M
306 CONTINUE
DO 350 I = IFT, ILA
TT = T(I, 1)
CALL CONVIT(T, XXHC, EMISS, XK)
QRI(I, 1) = XH * (TG - T(I, 1)) / XK
DTR(I, 1) = -QR(I, 1)
C
DTR2(I, 1) = 1. / 18. / DR(1) * 2 * (-85. * T(I, 1) + 108. * T(I, 2)
IF (MOD(NKONT, NP) .NE. 0) GO TO 40
PRINT 20, I, DTR(I, 1), DTR2(I, 1)
20 FORMAT(5H DTR(I, 1) =, E15.6, 10X, 7H DTR2 =, E15.6)
40 DTR(I, 2) = 1. / 6. / DR(1) * (-2. * T(I, 1) - 3. * T(I, 2) + 6. *
1 T(I, 3) - T(I, 4))
DTR2(I, 2) = 1. / DR(1) * 2 * (T(I, 3) - 2. * T(I, 2) + T(I, 1))
IF (MOD(NKONT, NP) .NE. 0) GO TO 41
PRINT 22, I, DTR(I, 2), DTR2(I, 2)
22 FORMAT(5H DTR(I, 2) =, E15.6, 10X, 7H DTR2 =, E15.6)
41 JJ1 = JJ5(I) - 2
DO 310 J = 3, JJ1
DTR(I, J) = 1. / 12. / DR(1) * (T(I, J-2) - 8. * T(I, J-1) + 8. *
1 T(I, J)) - T(I, J+1)
DTR2(I, J) = 1. / 12. / DR(1) * 2 * (-T(I, J-2) + 16. * T(I, J-1) -
IF (MOD(NKONT, NP) .NE. 0) GO TO 310
PRINT 24, I, J, DTR(I, J), DTR2(I, J)
310 CONTINUE
24 FORMAT(5H DTR(I, J) =, E15.6, 10X, 7H DTR2 =, E15.6)
DTR(I, JJ1+1) = 1. / 6. / DR(1) * (-6. * T(I, JJ1) + 3. * T(I, JJ1+1)
1 + 2. * T(I, JJ1+2) + T(I, JJ1-1))
DTR2(I, JJ1+1) = 1. / DR(1) * 2 * (T(I, JJ1+2) - 2. * T(I, JJ1+1) + T(I
1, JJ1))
IF (MOD(NKONT, NP) .NE. 0) GO TO 43
JZ = JJ1 + 1
43 IF(JJS(I) .EQ. J) GO TO 325
JJ = JJS(I)
DTRI(JJ) = 1.0 / (DR(2) / (1.0 + DR(2) / DR(1))) * (TI(J,JJ+1) - TI(J,JJ))
IDR2(JJ) = 2.0 / (DR(2) / DR(1)) * (TI(J,J+1) - (DR(2) / DR(1) + 1) * TI(J,JJJ) + TI(J,J + 1))
IF(MOD(NKONT,NP) .NE. 0) GO TO 44
PRINT 24, I, JJ, DTR1(JJ), IDR2(JJ)

44 JSX = 0
DO 319 JK = 2, JR
JSX = JSX + JJS(JK-1)
JP = JSX + 1
DTRJ(JP) = 1.0 / (DR(JK)* (TI(JP-1) - 8.0 * (TI(J,J-1) + 8.0 * TI(J,J)-1 - TI(J,J+1))
IDR2(JP) = 1.0 / (DR(JK)) * (TI(JP+1) - TI(JP-2))
IF(MOD(NKONT,NP) .NE. 0) GO TO 312
PRINT 24, I, JP, DTR(JP), IDR2(JP)

45JI = JP + 1
JL = JI + JJS(JK-1) - 4
DO 312 J = JI, JL
DTRJ(J) = 1.0 / (DR(JK)* (TI(J-2) - 8.0 * (TI(J,J-1) + 8.0 * TI(J,J)-1 - TI(J,J+1))
IDR2(J) = 1.0 / (DR(JK)) * (TI(JP+1) - TI(JP-2))
IF(MOD(NKONT,NP) .NE. 0) GO TO 319
PRINT 24, I, J, DTR(I,J), IDR2(I,J)

312 CONTINUE
JPP = JL + 1
DTRJ(JPP) = 1.0 / (DR(JK)* (TI(JPP-1) + 3.0 * (TI(J,J)-1 + 3.0 * (TI(J,J)+1 - TI(J,J+1)
IDR2(JPP) = 1.0 / (DR(JK)) * (TI(JPP+1) - 2.0 * (TI(JPP) + TI(J,J)+1))
IF(MOD(NKONT,NP) .NE. 0) GO TO 319
PRINT 24, I, JPP, DTR(JPP), IDR2(JPP)

47 IF(JK .EQ. JK) GO TO 315
JPP = JPP + 1
JZ = JK + 1
DTRJ(JZ) = 1.0 / (DR(JZ)/ (1.0 + DR(JZ)/ DR(JZ)) * (TI(JZ+1) - TI(J,1) * (TI(J,J)-1 + 2.0 * (DR(JZ)) - 2) * DR(JZ))
IDR2(JZ) = 1.0 / (DR(JZ)) * (TI(J,J)-1 - 2) * (DR(JZ))
IF(MOD(NKONT,NP) .NE. 0) GO TO 319
PRINT 24, I, JZ, DTR(JZ), IDR2(JZ)

315 CONTINUE
325 JLK = JSUM(JK) - 2
DTRJ(JLT+2) = 0.0
DTR2(JLT+2) = 2.0 / (DR(JL) * (TI(J,JLT+2) - 27.0 * (TI(J,JLT+1) - 11.0 / JLK))
IF(MOD(NKONT,NP) .NE. 0) GO TO 330
PRINT 24, I, JLK, DTR(JL), IDR2(JL)

330 CONTINUE
340 IF(IK .EQ. N) GO TO 360
IUN = UK + 1
IFT = IFT + LLS(IK)
ILA = ILA + LLS(IUN)
SUBROUTINE DKDT(TT, XK, DKT, XRHO, XCP, X, Y)
DIMENSION XI(20), YI(20)
CALL XKKSITT(XK, DKT)
XRHO = 490.0
I = 1
CALL LINEARITT(X, Y, XCP, I)
RETURN
END

SUBROUTINE XKKSITT(XK, DKT)
IF(TT .GT. 1472.0) GO TO 14
XK = 28.30 - .0087*TT
DKT = - 0.0087
GO TO 20
14 XK = 10.39 - .00347*TT
DKT = - 0.00347
20 CONTINUE
RETURN
END

SUBROUTINE LINEARITT(A, X, Y, VV, I)
DIMENSION XI(20), YI(20)
1 IF(Y(I+1) .LT. Y(I)) GO TO 10
C USE FOLLOWING IF AS Y INCREASES X INCREASES
10 IF(A-X(I+1))3, 2, 2
C USE FOLLOWING IF AS Y INCREASES X DECREASES
100 IF(A-X(I))2, 1, 2
GO TO 1
3 I = I - 1
VV = Y(I)*A-X(I+1)/X(I)-X(I+1)*Y(I+1)/(A-X(I))/(X(I+1)-X(I))
RETURN
END

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>.1714E-08</th>
<th>70.0</th>
<th>20.0</th>
<th>99550</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>.052083</td>
<td>.070325</td>
<td>.078125</td>
<td>.09896</td>
<td>.08</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1.0417</td>
<td>1.81907</td>
<td>3.50</td>
<td>.185</td>
<td>.180</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>.108</td>
<td>1.0</td>
<td>.12</td>
<td>4.0</td>
<td>.132</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>.15</td>
<td>1.0</td>
<td>.16</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>800.0</td>
<td>.15</td>
<td>1.0</td>
<td>.16</td>
<td>1.0</td>
<td>1.0</td>
<td>.180</td>
</tr>
<tr>
<td>Iteration Number</td>
<td>Axial Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.7000E-02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>0.1083E 02</td>
<td>ITERATION NUMBER 4000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 1</td>
<td>0.4254E 03</td>
<td>0.6798E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 2</td>
<td>0.4697E 03</td>
<td>0.5098E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 3</td>
<td>0.3100E 03</td>
<td>0.2604E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 4</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 5</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 6</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 7</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 8</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 9</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 10</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 11</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 12</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 13</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 14</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 15</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 16</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 17</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 18</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 19</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 20</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 21</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 22</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 23</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 24</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 25</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 26</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 27</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 28</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 29</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 30</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 31</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 32</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 33</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 34</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 35</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 36</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 37</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 38</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 39</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXIAL LOCATION 40</td>
<td>0.2694E 03</td>
<td>0.2596E 03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

31
APPENDIX B

FLOW CHART OF MAIN PROGRAM
FLOW CHART OF MAIN PROGRAM

Start

Read in Data

Compute Time Interval

CALL QRSUB

CALL QZSUB

CALL RADDR

1

1

CALL AXIDER

CALL DKDT

Compute Temperature Change

PRINT OUT Computed Data

If desired time is reached CALL EXIT, otherwise go back to 1 and continue
LIST OF SYMBOLS USED IN TEXT

Symbols

- **C** - specific heat (BTU/lb °F)
- **h** - heat transfer coefficient (BTU/hr ft² °F)
- **i** - number of radial nodes
- **j** - number of axial nodes
- **k** - thermal conductivity (BTU/hr ft °F)
- **m** - axial node
- **n** - radial node
- **q** - heat flux (BTU/hr ft²)
- **R** - radial boundary (ft)
- **r** - radial coordinate (ft)
- **T** - temperature (°F)
- **t** - time (hr)
- **Z** - axial coordinate (ft)
- **ΔR** - radial increment (ft)
- **ΔZ** - axial increment (ft)
- **ε** - emissivity
- **ρ** - density (lb/ft³)
- **σ** - radiation coefficient (0.1714 BTU/hr ft² °R⁴)

Subscripts

- **g** - gas
- **i** - initial value
- **m** - node m
- **n** - node n
- **0** - surroundings
- **1** - boundary 1, segment 1
- **2** - boundary 2, segment 2
- **3** - boundary 3, segment 3
- **4** - boundary 4