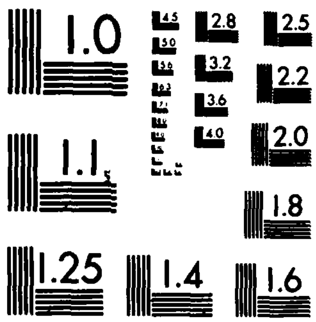
AD-A128 793 COMPUTER MODELS FOR TWO-DIMENSIONAL STEADY-STATE HEAT CONDUCTION(U) COLD REGIONS RESEARCH AND ENGINEERING LAB HANOVER NH M R ALBERT ET AL. APR 83 CRREL-83-10 1/1

UNCLASSIFIED F/G 9/2 NL

END
DATE
FILMED
DTIC




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

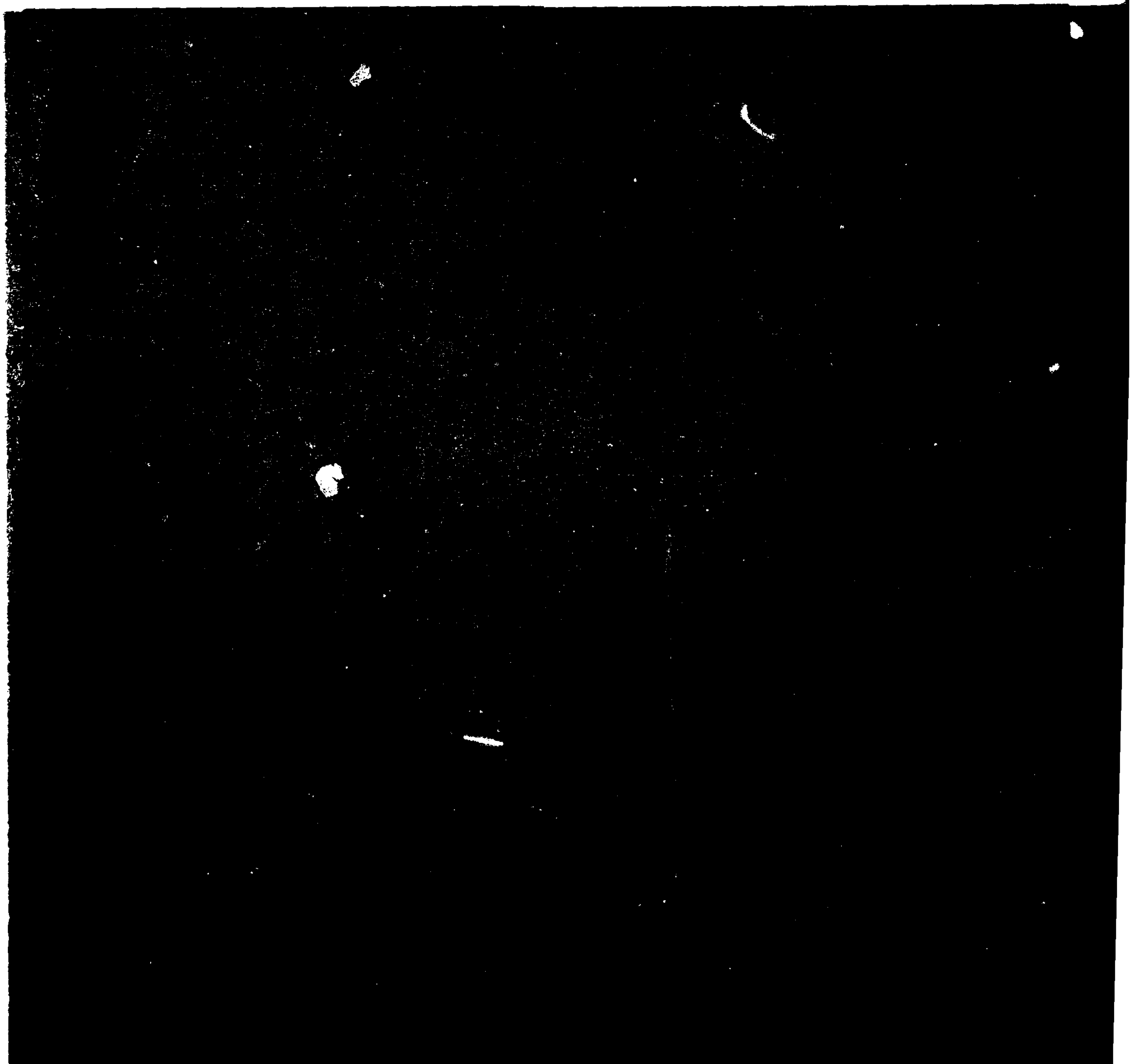
# CRREL
# REPORT 83-10

US Army Corps of Engineers
Cold Regions Research & Engineering Laboratory

(12)

AD A128 793

# *Computer models for two-dimensional steady-state heat conduction*



*For conversion of SI metric units to U.S./British customary units of measurement consult ASTM Standard E380,* Metric Practice Guide, *published by the American Society for Testing and Materials, 1916 Race St., Philadelphia, Pa. 19103.*

Cover: *A circular boundary modeled by the finite difference method (left) and the finite element method (right).*

# CRREL Report 83-10

April 1983

# *Computer models for two-dimensional steady-state heat conduction*

M.R. Albert and G. Phetteplace



Prepared for
OFFICE OF THE CHIEF OF ENGINEERS

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
CRREL Report 83-10

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
COMPUTER MODELS FOR TWO-DIMENSIONAL STEADY-STATE HEAT CONDUCTION

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
M.R. Albert and G. Phetteplace

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
U.S. Army Cold Regions Research and Engineering Laboratory
Hanover, New Hampshire 03755

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
DA Project 4A762730AT42
Technical Area D, Work Unit 017

11. CONTROLLING OFFICE NAME AND ADDRESS
Office of the Chief of Engineers
Washington, D.C. 20314

12. REPORT DATE
April 1983

13. NUMBER OF PAGES
90

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Approved for public release; distribution unlimited.

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer program documentation
Computer programs
Heat conduction
Heat transfer
Mathematical models

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report outlines the development and verification of two computer models of two-dimensional steady-state heat conduction, including a variety of boundary conditions. One is a finite difference program and the other is a finite element program. The results of each program are compared to two analytic solutions, and to one another. Sample input and output, and user instructions for each program are included. The programs have the same accuracy when modeling problems involving rectangular boundaries. The finite element method is better able to model curved boundaries, while the finite difference program is easier to use.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

## PREFACE

This report was prepared by Mary Remley Albert, Mathematician, and Gary E. Phetteplace, Mechanical Engineer, both of the Applied Research Branch, Experimental Engineering Division, U.S. Army Cold Regions Research and Engineering Laboratory. The work was sponsored by DA Project 4A762730AT42, *Design, Construction and Operations Technology for Cold Regions,* Technical Area D, *Cold Regions Design and Construction,* Work Unit 017, *Heat Distribution Systems in Cold Regions.*

The authors thank Professor Francis Kennedy of Dartmouth College and Dr. Kevin O'Neill of CRREL for their technical review of this report.

The contents of this report are not to be used for advertising or promotional purposes. Citation of brand names does not constitute an official endorsement or approval of the use of such commercial products.

## CONTENTS



## ILLUSTRATIONS

## TABLES

# COMPUTER MODELS FOR TWO-DIMENSIONAL STEADY-STATE HEAT CONDUCTION

M.R. Albert and G. Phetteplace

## INTRODUCTION

Most major Army facilities are heated by central heat distribution systems. Heat losses from these distribution systems are of interest, as are temperatures of soil surrounding the distribution pipes, for several reasons. If we are to design efficient heat distribution systems we must be able to accurately assess the heat losses in order to determine the optimum trade-off between the cost of insulation and the continuing cost of heat loss. In instances where portions of a system have failed and require replacement, again we need to accurately assess potential heat losses to determine the optimum insulation thickness for the areas that must be replaced. In areas of seasonal frost and permafrost, the temperature distribution in the soil around buried heat carrying pipes is of primary importance. Significant thawing of permafrost can cause loss of support and subsequent pipe settlement and possible pipe breakage. In areas of seasonal frost, piping systems should be buried deep enough to prevent freezeup in the event of a system outage.

Several methods exist for analyzing heat transfer problems of this type. Some of the methods which have been applied to conduction heat transfer include: 1) analytical methods, 2) approximate methods, 3) empirical methods, and 4) numerical methods.

Analytical methods are useful for a limited class of problems for which closed form analytical solutions to the heat conduction equation have been found. They are limited because only the simplest geometries may be treated with corresponding ideal boundary conditions. One such solution exists for a buried uninsulated single pipe in a semi-infinite medium with uniform and constant material properties and boundary conditions; this will be discussed later. Because of the very limited class of problems for which closed form analytical solutions have been found, they have little application to real world engineering problems. However, they can be used to find approximate solutions to actual problems.

Approximate methods are actually an extension of analytical methods. In some instances an exact solution to the heat conduction equation cannot be found, yet by making some reasonable assumptions an approximate solution may become possible. Approximate solutions are limited to certain problems, like analytical solutions, except to a lesser extent. They can provide estimates suitable for applications not requiring precise results.

Empirical methods are based on experimental results instead of on the solution of the governing equation. With knowledge of the form of the governing equation and solution, experimental data can be used to find empirical equations which approximate the physical process. A typical example of this approach is the development of conduction shape factors by electrical analog experiments. Again, empirical methods are limited to cases where empirical equations have been found. These equations are usually only applicable to very specific problems and often cannot be accurately extrapolated to describe similar problems.

Numerical methods have found many applications in heat transfer. In short, a numerical method uses an approximation of the governing differential equation, or its solution, at a number of discrete intervals of a region to find an approximate solution to the heat conduction equation over that discretized region. The enormous advantage of numerical methods is their ability to model nearly any problem, including ones without ideal boundary conditions. This is an advantage no other method can claim. The disadvantage of numberical methods lies in their complexity. Criteria governing the stability, convergence, consistency, and accuracy of a numerical method must be established before the method may be considered valid, and for anything but the simplest problems a computer is required for solutions. Fortunately, however, the computer program can be written to handle a broad class of problems and the user need only be familiar with certain aspects of the problem to accurately use the program to find the results.

The purpose of this report is to outline the finite difference and finite element numerical methods as they apply to steady-state heat conduction, to compare the results of each method to analytical results and also to the results of the other method, and to demonstrate the use of each method's computer program to the potential user.

## THE FINITE DIFFERENCE METHOD

### Basic ideas of finite differences

The finite difference method is a numerical method used to approximate the solution of a partial differential equation. Common partial differential equations for which finite differences have been employed include the wave equation, the vorticity equation, Poisson's equation, and the heat conduction equation,

$$\frac{\partial}{\partial y}\left(k\,\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial x}\left(k\,\frac{\partial T}{\partial x}\right) = \rho C_p\,\frac{\partial T}{\partial t}\ .$$

Often boundary conditions or nonlinearities in a problem involving differential equations complicate the problem so that analytical solutions are unavailable and numerical solutions must be used. The finite difference method approximates each partial derivative in the differential equation by an algebraic finite difference representation.

Finite difference representations may be derived either from a Taylor series expansion about a point or from physical considerations, such as an energy balance in the case of heat conduction. Let us briefly review the Taylor series procedure. Consider the forward Taylor expansion of a function $f(x)$ about $x$:

$$f(x+\Delta x) = f(x) + \Delta x f'(x) + \frac{(\Delta x)^2}{2}\,f''(x) + \frac{(\Delta x)^3}{6}\,f'''(x) + \ldots \qquad (1)$$

where the primes denote differentiation with respect to $x$. This may be solved for $f'(x)$ by truncating the higher-order derivatives to obtain

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x)\ . \qquad (2)$$

This is the forward finite difference expression for the first derivative of the function $f(x)$. The term $O(\Delta x)$ indicates that the error due to truncation of the Taylor series is of the order $\Delta x$.

The representation is commonly used, for example, as an approximation of $\partial T/\partial t$ in the heat conduction equation, where we let

$$\frac{\partial T}{\partial t} = \frac{T(t + \Delta t) - T(t)}{\Delta t} \tag{3}$$

where

$T(t)$ = temperature at time $t$

$\Delta t$ = time step.

Similarly, we may find the backward finite difference expression for the first derivative

$$f(x - \Delta x) = f(x) - (\Delta x) f'(x) + \frac{(\Delta x)^2}{2} f''(x) - \frac{(\Delta x)^3}{6} f'''(x) + \ldots \tag{4}$$

where

$$f'(x) = \frac{f(x) - f(x - \Delta x)}{\Delta x} + O(\Delta x) .$$

To obtain a difference representation for the second derivative, eq 1 and 4 are added

$$f(x + \Delta x) + f(x - \Delta x) = 2f(x) + (\Delta x)^2 f''(x) + O[(\Delta x)^2]$$

$$f''(x) = \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{(\Delta x)^2} . \tag{5}$$

Equation 5 is the central difference representation for the second derivative. Note that it is accurate to the order of $(\Delta x)^2$. It is evident that the accuracy of a solution found using finite differences will be dependent upon our taking $\Delta x$ sufficiently small.

When using the finite difference method, we set up a network of grid points, called "nodes," in the region of the spatial independent variables in which we are interested. The boundaries of the grid should coincide with those of the problem. The partial derivatives of the original partial differential equation are replaced by their corresponding difference representations, and an equation is set up for each point in the grid. The resulting set of algebraic equations may then be solved.

**Finite difference computer program**

A finite difference computer program was written to model steady-state two-dimensional heat conduction. The program has been set up in a general form to allow many different conductivities in the region of interest and to permit constant flux, convective, constant temperature, and semi-infinite boundary conditions. The general form of the program allows new problems to be set up very easily.

**Application of finite differences to steady-state heat conduction**

For two-dimensional steady-state heat conduction, heat transfer obeys the elliptic partial differential equation

$$\frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k \frac{\partial T}{\partial y}\right) = 0 \tag{6}$$

where

$T$ = temperature

$k$ = thermal conductivity

$x,y$ = spatial variables.

If the conductivity $k$ were to be constant over the region, the finite difference form for $\partial/\partial x(k\partial T/\partial x) = k\partial^2 T/\partial x^2$ would be immediately obtainable from eq 5,

$$k \frac{\partial^2 T}{\partial x^2} = k \frac{T(x+\Delta x) - 2T(x) + T(x-\Delta x)}{(\Delta x)^2} .$$

However, the finite difference computer program was to be written so that conductivity could be a function of location. Then the resultant conductivity must be used in regions of varying conductivity. The "resultant conductivity" is easiest to understand by recalling that conductivity is the reciprocal of resistance. The net resistance between any node $x, y$ and node $x+\Delta x, y$ is

$$R = R_x + R_{x+\Delta x} = \frac{d_x}{k_x} + \frac{d_{x+\Delta x}}{k_{x+\Delta x}}$$

where, for example, $d_x$ is the fraction of the distance along the heat flow path which is associated with conductivity $k_x$. In this program, space increments $\Delta x$ and $\Delta y$ are assumed to be uniform, then $d_x = d_{x+\Delta x} = ½$. The resultant conductivity is the reciprocal of resistance $R$:

$$\frac{1}{R} = \frac{2}{\frac{1}{k_x} + \frac{1}{k_{x+\Delta x}}} .$$

Thus the finite difference formulation for each term in eq 6 is as follows:

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) = \left(\frac{2}{\frac{1}{k_{x+\Delta x}} + \frac{1}{k_x}}\right) \frac{T(x+\Delta x) - T(x)}{(\Delta x)^2} + \left(\frac{2}{\frac{1}{k_{x-\Delta x}} + \frac{1}{k_x}}\right) \frac{T(x-\Delta x) - T(x)}{(\Delta x)^2} \qquad (7)$$

$$\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) = \left(\frac{2}{\frac{1}{k_{y+\Delta y}} + \frac{1}{k_y}}\right) \frac{T(y+\Delta y) - T(y)}{(\Delta y)^2} + \left(\frac{2}{\frac{1}{k_{y-\Delta y}} + \frac{1}{k_y}}\right) \frac{T(y-\Delta y) - T(y)}{(\Delta y)^2} . \qquad (8)$$

Combining eq 7 and 8, we arrive at the finite difference formulation for eq 6

$$\left(\frac{2}{\frac{1}{k_{x+\Delta y}} + \frac{1}{k_x}}\right) \frac{T_{x+\Delta x,y} - T_{x,y}}{(\Delta x)^2} + \left(\frac{2}{\frac{1}{k_{x-\Delta x}} + \frac{1}{k_x}}\right) \frac{T_{x-\Delta x,y} - T_{x,y}}{(\Delta x)^2} +$$

$$\left(\frac{2}{\frac{1}{k_{y+\Delta y}} + \frac{1}{k_y}}\right) \frac{T_{x,y+\Delta y} - T_{x,y}}{(\Delta y)^2} + \left(\frac{2}{\frac{1}{k_{y-\Delta y}} + \frac{1}{k_y}}\right) \frac{T_{x,y-\Delta y} - T_{x,y}}{(\Delta y)^2} = 0 . \qquad (9)$$

Here it is convenient to change the notation to a more commonly used form. Let $T_{i,j}$ represent the temperature of the node under consideration, where $i$ is increased in the vertical direction and $j$ in the horizontal direction. The grid set up will appear as shown in Figure 1 (the length and width of the grid are variable, and may be established when data are entered into the computer program).

Each node represents the area (enclosed in dotted lines in Fig. 1) around it. The most convenient way to set up the computer program to enable it to handle many different cases is to use a square grid, that is, let $\Delta x = \Delta y = \Delta s$. Now we multiply each term in eq 9 by $(\Delta s)^2$, change to the new notation, and rearrange the terms to arrive at the usual finite difference equation for a node $i,j$ not on a boundary,

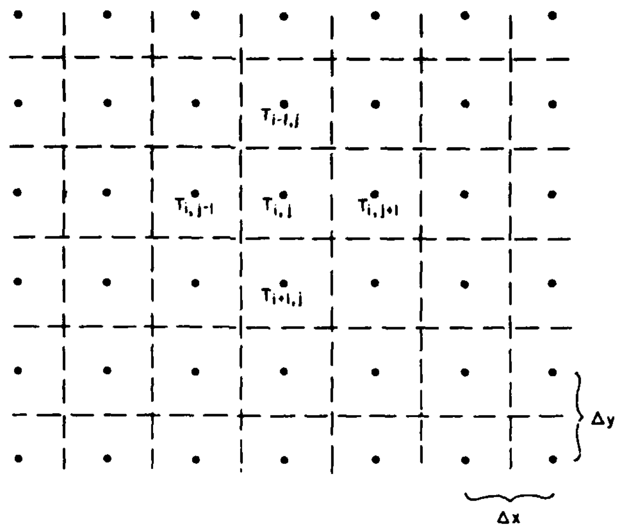



*Figure 1. Finite difference grid (boundaries not yet specified).*

$$
\begin{aligned}
&\left(\frac{2}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right) T_{i,j-1}+\left(\frac{2}{\frac{1}{k_{i,j+1}}+\frac{1}{k_{i,j}}}\right) T_{i,j+1}+\left(\frac{2}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right) T_{i+1,j} \\
&+\left(\frac{2}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}\right) T_{i-1,j}-\left(\frac{2}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}+\frac{2}{\frac{1}{k_{i,j+1}}+\frac{1}{k_{i,j}}}\right. \\
&\left.+\frac{2}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}+\frac{2}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right) T_{i,j}=0 . \qquad (10)
\end{aligned}
$$

Note that each node represents a region of space, and will therefore have a specified thermal conductivity. The above equation allows for a different conductivity for each node of the grid.

**Boundary conditions**

The finite difference equations used in heat transfer may be derived either by replacement of a partial derivative by its difference representation, as above, or by calculating an energy balance between a node and each of its surrounding nodes. The boundary conditions in the following sections will be derived using energy balance considerations.

*Constant flux boundary*

For a node on a constant flux boundary, the condition $dT/dx = C$ exists at the boundary. Consider a node on the right-hand boundary. The control volume is that which is associated with the node, as illustrated in Figure 2. Examine the heat flow between node $i,j$ and its surrounding nodes. Between node $i,j$ and node $i$-1,$j$ we have, for unit depth,

$$Q_1=\left(\frac{2}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}\right) \frac{\Delta x}{2} \frac{1}{\Delta y}\left(T_{i-1,j}-T_{i,j}\right)$$

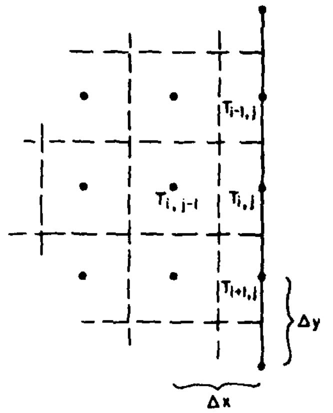



*Figure 2. Node on boundary of finite difference grid.*

where

$$\frac{2}{\frac{1}{k_{i-1,j}} + \frac{1}{k_{i,j}}} = \text{resultant thermal conductivity between nodes } i\text{-}1,j \text{ and } i,j$$

$$\frac{\Delta x}{2} = \text{surface area perpendicular to the direction of heat flow, for unit depth}$$

$$\Delta y = \text{distance between the nodes}$$

$$T_{i-1,j} - T_{i,j} = \text{temperature difference between nodes.}$$

Similarly, for the heat flow from node $i$+1,$j$ into node $i,j$ we have

$$Q_3 = \left(\frac{2}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}}\right) \frac{\Delta x}{2\Delta y} \left(T_{i+1,j} - T_{i,j}\right),$$

and the flow from node $i,j$-1 into node $i,j$ is

$$Q_2 = \left(\frac{2}{\frac{1}{k_{i,j-1}} + \frac{1}{k_{i,j}}}\right) \frac{\Delta y}{\Delta x} \left(T_{i,j-1} - T_{i,j}\right).$$

The heat flow crossing the boundary into node $i,j$ is given by

$$Q_4 = \phi \cdot \Delta s$$

where $\phi$ is the heat flux crossing the boundary per unit area. At steady state the sum of the heat flows is zero. Then we have

$$Q_1 + Q_2 + Q_3 + Q_4 = 0.$$

For $\Delta x = \Delta y = \Delta s$,

$$\left(\frac{1}{\frac{1}{k_{i-1,j}} + \frac{1}{k_{i,j}}}\right) T_{i-1,j} + \left(\frac{2}{\frac{1}{k_{i,j-1}} + \frac{1}{k_{i,j}}}\right) T_{i,j-1} + \left(\frac{1}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i+1,j}}}\right) T_{i+1,j}$$

$$-\left(\frac{1}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}+\frac{2}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}+\frac{1}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right)T_{i,j}=-\phi\Delta s\,. \tag{11}$$

This equation applies to the right-hand boundar ; the indices may be appropriately rearranged to obtain the equation for other boundaries. Note that, for a boundary which is insulated or on a line of symmetry, the zero heat flux condition holds and $\phi = 0$.

*Convective boundary*

Again consider the boundary illustrated in Figure 2. When surface convection is present, the heat flow at the boundary is $Q = h\Delta y(T_A - T_{i,j})$, where $T_A$ is the ambient temperature outside the grid, and $h$ is the surface heat transfer coefficient. The heat flow between node $i, j$ and the surrounding three nodes will be the same as that given for the constant flux boundary. The steady-state formulation for the right-hand boundary with $\Delta x = \Delta y = \Delta s$ is

$$\left(\frac{1}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}\right)(T_{i-1,j}-T_{i,j})+\left(\frac{2}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right)(T_{i,j-1}-T_{i,j})$$

$$+\left(\frac{1}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right)(T_{i+1,j}-T_{i,j})+h\Delta s(T_A-T_{i,j})=0\,. \tag{12}$$

and rearranging gives,

$$\left(\frac{1}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}\right)T_{i-1,j}+\left(\frac{2}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right)T_{i,j-1}$$

$$+\left(\frac{1}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right)T_{i+1,j}-\left(\frac{1}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}+\frac{1}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right.$$

$$\left.+\frac{1}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}+h\Delta s\right)T_{i,j}=-h\Delta sT_A\,. \tag{13}$$

Again, the indices may be suitably rearranged for nodes on other boundaries.

*Constant temperature boundary*

For a constant temperature node on any boundary or inside the gride we have $T_{i,j} = C$, where $C$ is the temperature of the node.

*Semi-infinite boundary*

The semi-infinite boundary condition represents a continuous, uniform material extending ad infinitum in one direction, with a known temperature at infinity. This boundary is approximated in the finite difference computer program by specifying a large distance between the boundary node and the node adjacent to it, and the condition requires a special heat balance for nodes adjacent to the boundary as well as for the boundary nodes. For example, consider the right-hand semi-infinite boundary illustrated in Figure 3, on the edge of a grid with uniform internodal

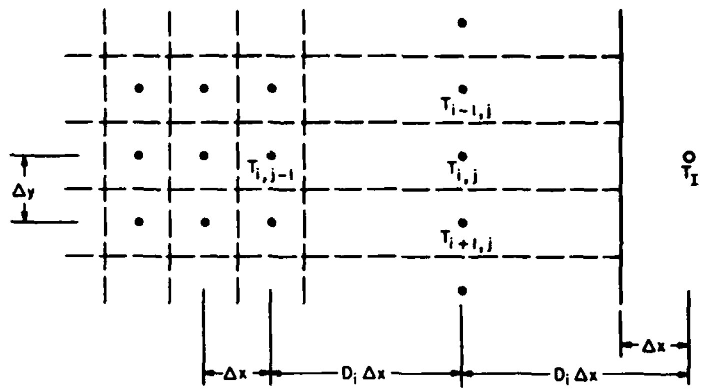



*Figure 3. Semi-infinite boundary approximation.*

distances. The temperature at infinity is represented by an imaginary node $I$, located outside the grid. $D_i \cdot \Delta x$ is the distance from the internal node to the node on the boundary, and also the distance from the boundary node to the imaginary node at infinity. As previously stated, the accuracy of the calculated temperature distribution is dependent on the distance between nodes; thus when choosing $D_i$, we should choose the smallest distance which may still be considered large in terms of the grid size.

The heat flow between node $i, j$ and each of the surrounding nodes is as follows:

$$Q_1 = \left( \frac{2}{\frac{1}{k_{i-1,j}} + \frac{1}{k_{i,j}}} \right) (2D_i - 1)\,\Delta x \cdot \frac{1}{\Delta y} \quad (T_{i-1,j} - T_{i,j}),$$

$$Q_2 = \left( \frac{2D_i}{\frac{1}{k_{i,j-1}} + \frac{2D_i - 1}{k_{i,j}}} \right) \Delta y \cdot \frac{1}{D_i\,\Delta x} \quad (T_{i,j-1} - T_{i,j}),$$

$$Q_3 = \left( \frac{2}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} \right) (2D_i - 1)\,\Delta x \cdot \frac{1}{\Delta y} \quad (T_{i+1,j} - T_{i,j}),$$

$$Q_4 = \left( \frac{2D_i}{\frac{1}{k_I} + \frac{2D_i - 1}{k_{i,j}}} \right) \Delta y \cdot \frac{1}{D_i\,\Delta x} \quad (T_I - T_{i,j}) .$$

For steady state, the heat balance yields $Q_1 + Q_2 + Q_3 + Q_4 = 0$. Allowing $\Delta x = \Delta y$, we find

$$\left( \frac{2(2D_i - 1)}{\frac{1}{k_{i-1,j}} + \frac{1}{k_{i,j}}} \right) T_{i-1,j} + \left( \frac{2}{\frac{1}{k_{i,j-1}} + \frac{2D_i - 1}{k_{i,j}}} \right) T_{i,j-1}$$

$$+ \left( \frac{2(2D_i - 1)}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} \right) T_{i+1,j} - \left( \frac{2(2D_i - 1)}{\frac{1}{k_{i-1,j}} + \frac{1}{k_{i,j}}} + \frac{2}{\frac{1}{k_{i,j-1}} + \frac{2D_i - 1}{k_{i,j}}} + \right.$$

$$+\frac{2(2D_i-1)}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}+\frac{2}{\frac{1}{k_i}+\frac{2D_i-1}{k_{i,j}}}\Bigg)\ T_{i,j}=-\frac{2T_i}{\frac{1}{k_i}+\frac{2D_i-1}{k_{i,j}}}\ . \qquad (14)$$

Now consider the heat flow for the square node adjacent to the irregular semi-infinite node, as illustrated in Figure 4.

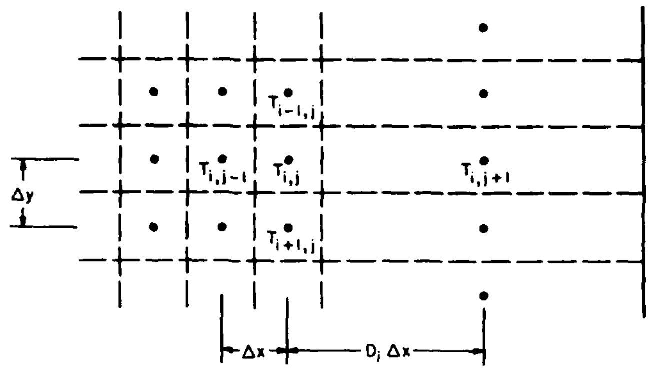



*Figure 4. Node adjacent to semi-infinite boundary node.*

The heat flow between the node and each of its adjacent nodes is

$$Q_1=\left(\frac{2}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}\right)\ \Delta x\cdot\frac{1}{\Delta y}\ (T_{i-1,j}-T_{i,j}),$$

$$Q_2=\left(\frac{2}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right)\ \Delta y\cdot\frac{1}{\Delta x}\ (T_{i,j-1}-T_{i,j}),$$

$$Q_3=\left(\frac{2}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right)\ \Delta x\cdot\frac{1}{\Delta y}\ (T_{i+1,j}-T_{i,j}),$$

$$Q_4=\left(\frac{2D_i}{\frac{2D_i-1}{k_{i,j+1}}+\frac{1}{k_{i,j}}}\right)\ \Delta y\cdot\frac{1}{D_i\Delta x}\ (T_{i,j+1}-T_{i,j})\ .$$

The resulting equation for steady state, for $\Delta x=\Delta y$, is

$$\left(\frac{2}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}\right)T_{i-1,j}+\left(\frac{2}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right)T_{i,j-1}+\left(\frac{2}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right)T_{i+1,j}$$

$$+\left(\frac{2}{\frac{2D_i-1}{k_{i,j+1}}+\frac{1}{k_{i,j}}}\right)T_{i,j+1}-\left(\frac{2}{\frac{1}{k_{i-1,j}}+\frac{1}{k_{i,j}}}+\frac{2}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}+\right.$$

$$+ \frac{2}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} + \left( \frac{2}{\frac{2D_i - 1}{k_{i,j+1}} + \frac{1}{k_{i,j}}} \right) T_{i,j} = 0 \tag{15}$$

*Constant flux corner*

Consider the corner shown in Figure 5, subject to the constant flux condition on two sides,

$$Q_1 = \left( \frac{2}{\frac{1}{k_{i,j-1}} + \frac{1}{k_{i,j}}} \right) \frac{\Delta y}{2\Delta x} (T_{i,j-1} - T_{i,j})$$

$$Q_2 = \left( \frac{2}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} \right) \frac{\Delta x}{2\Delta y} (T_{i+1,j} - T_{i,j})$$

$$Q_3 = \tfrac{1}{2} (\phi_T + \phi_S) \Delta s .$$

$\phi_T$ is the heat flux per unit area crossing the top boundary, $\phi_S$ is that crossing the side. For steady state, $Q_1 + Q_2 + Q_3 = 0$. In general, for $\Delta x = \Delta y = \Delta s$,

$$\left( \frac{1}{\frac{1}{k_{i,j-1}} + \frac{1}{k_{i,j}}} \right) T_{i,j-1} + \left( \frac{1}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} \right) T_{i+1,j}$$

$$- \left( \frac{1}{\frac{1}{k_{i,j-1}} + \frac{1}{k_{i,j}}} + \frac{1}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} \right) T_{i,j} = -(\phi_T + \phi_S) \frac{\Delta s}{2} . \tag{16}$$

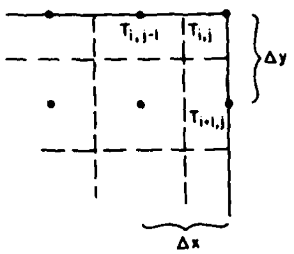



*Figure 5. Node on corner of finite difference grid.*

*Convective corner*

For the corner shown in Figure 5, subject to convection on both sides, convection along the top surface yields

$$Q_4 = h \frac{\Delta x}{2} (T_A - T_{i,j}),$$

and along the right side,

$$Q_3 = h \frac{\Delta y}{2} (T_A - T_{i,j}) .$$

With $Q_1$ and $Q_2$ the same as for the previous corner, and for $\Delta x = \Delta y = \Delta s$, the equation for steady state is as follows:

$$\left(\frac{1}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right) T_{i,j-1} + \left(\frac{1}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right) T_{i+1,j}$$

$$-\left(\frac{1}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}} + \frac{1}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}} + h\Delta s\right) T_{i,j} = -h\Delta s T_A \,. \tag{17}$$

*Semi-infinite corner*

Consider the node pictured in Figure 6, on the corner between two semi-infinite boundaries. $T_{l_r}$ is the temperature a large distance to the right of the grid, and $T_{l_t}$ is the temperature away from the top of the grid.

$$Q_1 = \left(\frac{2D_i}{\frac{1}{k_{l_t}}+\frac{2D_i-1}{k_{i,j}}}\right) (2D_i-1)\,\Delta x \cdot \frac{1}{D_i \cdot \Delta y} \left(T_{l_t} - T_{i,j}\right),$$

$$Q_2 = \left(\frac{2D_i}{\frac{1}{k_{i,j-1}}+\frac{2D_i-1}{k_{i,j}}}\right) (2D_i-1)\,\Delta y \cdot \frac{1}{D_i \cdot \Delta x} \left(T_{i,j-1} - T_{i,j}\right),$$

$$Q_3 = \left(\frac{2D_i}{\frac{1}{k_{i+1,j}}+\frac{2D_i-1}{k_{i,j}}}\right) (2D_i-1)\,\Delta x \cdot \frac{1}{D_i \Delta y} \left(T_{i+1,j} - T_{i,j}\right)$$

$$Q_4 = \left(\frac{2D_i}{\frac{1}{k_{l_r}}+\frac{2D_i-1}{k_{i,j}}}\right) (2D_i-1)\,\Delta y \, \frac{1}{D_i \Delta x} \left(T_{l_r} - T_{i,j}\right).$$

For steady state, the heat flows sum to zero. Summing the above, then multiplying each term by $1/(2D_i-1)$ and allowing $\Delta x = \Delta y$, gives us the heat balance for this corner node

$$\left(\frac{2}{\frac{1}{k_{i,j-1}}+\frac{2D_i-1}{k_{i,j}}}\right) T_{i,j-1} + \left(\frac{2}{\frac{1}{k_{i+1,j}}+\frac{2D_i-1}{k_{i,j}}}\right) T_{i+1,j} - \left(\frac{2}{\frac{1}{k_{l_t}}+\frac{2D_i-1}{k_{i,j}}}\right.$$

$$\left. + \frac{2}{\frac{1}{k_{i,j-1}}+\frac{2D_i-1}{k_{i,j}}} + \frac{2}{\frac{1}{k_{i+1,j}}+\frac{2D_i-1}{k_{i,j}}} + \frac{2}{\frac{1}{k_{l_r}}+\frac{2D_i-1}{k_{i,j}}}\right) T_{i,j}$$

$$= -\left(\frac{2}{\frac{1}{k_{l_t}}+\frac{2D_i-1}{k_{i,j}}}\right) T_{l_t} - \left(\frac{2}{\frac{1}{k_{l_r}}+\frac{2D_i-1}{k_{i,j}}}\right) T_{l_r} \,. \tag{18}$$

In the computer program, it is assumed that $k_{l_t} = k_{i,j} = k_{l_r}$.

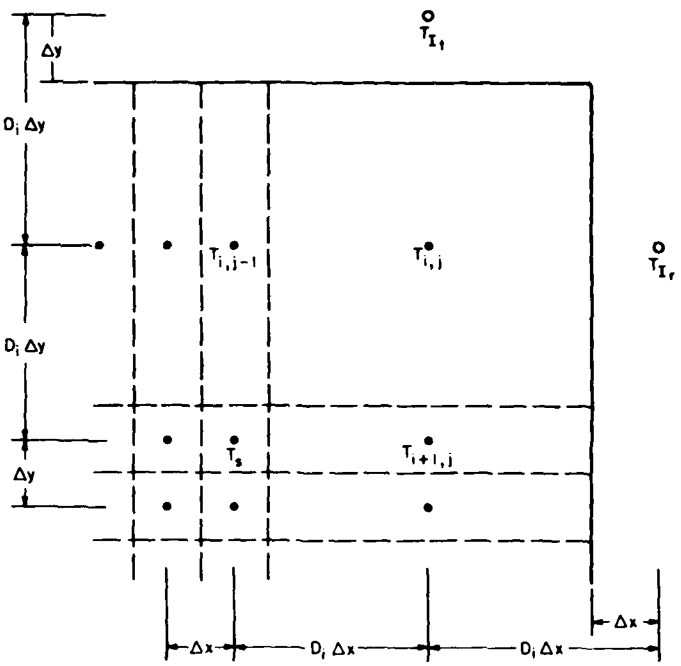



*Figure 6. Node on corner between two semi-infinite boundaries.*

When this semi-infinite condition is used for a corner, a special heat balance for the interior node labeled $T_S$ in Figure 6 must be used. This balance will be the same as that given by eq 15, except that $Q_1$ will be replaced by

$$Q_1 = \left( \frac{2D_i}{\frac{2D_i - 1}{k_{i-1,j}} + \frac{1}{k_{i,j}}} \right) \Delta x \frac{1}{D_i \Delta y} \left( T_{i-1,j} - T_{i,j} \right).$$

Then the full heat balance is given by

$$\left( \frac{2}{\frac{2D_i-1}{k_{i-1,j}} + \frac{1}{k_{i,j}}} \right) T_{i-1,j} + \left( \frac{2}{\frac{1}{k_{i,j-1}} + \frac{1}{k_{i,j}}} \right) T_{i,j-1} + \left( \frac{2}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} \right) T_{i+1,j}$$

$$+ \left( \frac{2}{\frac{2D_i-1}{k_{i,j+1}} + \frac{1}{k_{i,j}}} \right) T_{i,j+1} - \left( \frac{2}{\frac{2D_i-1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} + \frac{2}{\frac{1}{k_{i,j-1}} + \frac{1}{k_{i,j}}} \right.$$

$$\left. + \frac{2}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} + \frac{2}{\frac{2D_i-1}{k_{i,j+1}} + \frac{1}{k_{i,j}}} \right) T_{i,j} = 0 . \tag{19}$$

*Corner with one side constant flux, the other side convective*

It is possible to have a corner subject to convection on one side and constant flux on the other side. For the corner in Figure 5, subject to constant flux on the right side and convection on the surface,

$$Q_1 = \left(\frac{2}{\dfrac{1}{k_{i,j-1}} + \dfrac{1}{k_{i,j}}}\right) \frac{\Delta y}{2\Delta x} \quad (T_{i,j-1} - T_{i,j}) \, ,$$

$$Q_2 = \left(\frac{2}{\dfrac{1}{k_{i+1,j}} + \dfrac{1}{k_{i,j}}}\right) \frac{\Delta x}{2\Delta y} \quad (T_{i+1,j} - T_{i,j}) \, ,$$

$$Q_3 = \tfrac{1}{2} \, \phi \Delta s \, ,$$

$$Q_4 = h \, \frac{\Delta x}{2} \, (T_A - T_{i,j}) \, .$$

Let $\Delta x = \Delta y = \Delta s$,

$$\left(\frac{1}{\dfrac{1}{k_{i,j-1}} + \dfrac{1}{k_{i,j}}}\right) T_{i,j-1} + \left(\frac{1}{\dfrac{1}{k_{i+1,j}} + \dfrac{1}{k_{i,j}}}\right) T_{i+1,j} - \left(\frac{1}{\dfrac{1}{k_{i,j-1}} + \dfrac{1}{k_{i,j}}} + \frac{1}{\dfrac{1}{k_{i+1,j}} + \dfrac{1}{k_{i,j}}} + \tfrac{1}{2} h \Delta s\right) T_{i,j} = -\tfrac{1}{2} h \Delta s T_A - \tfrac{1}{2} \phi \Delta s \, . \tag{20}$$

*Corner with one side constant flux, the other side semi-infinite*

As illustrated in Figure 7, let the top right-hand corner of the grid have a semi-infinite boundary on the right side and a constant flux boundary on the top. Allow the heat flow per unit area crossing the top of the grid to be $\phi$. Then the heat flows for node $i, j$ are given by

$$Q_1 = \phi(2D_i - 1) \, \Delta x \, ,$$

$$Q_2 = \left(\frac{2D_i}{\dfrac{1}{k_{i,j-1}} + \dfrac{2D_i - 1}{k_{i,j}}}\right) \left(\frac{\Delta y}{2}\right) \left(\frac{1}{D_i \Delta x}\right) \quad (T_{i,j-1} - T_{i,j}) \, ,$$

$$Q_3 = \left(\frac{2}{\dfrac{1}{k_{i+1,j}} + \dfrac{1}{k_{i,j}}}\right) \quad (2D_i - 1) \, \Delta x \left(\frac{1}{\Delta y}\right) \quad (T_{i+1,j} - T_{i,j}) \, ,$$

$$Q_4 = \left(\frac{2D_i}{\dfrac{1}{k_I} + \dfrac{2D_i - 1}{k_{i,j}}}\right) \left(\frac{\Delta y}{2}\right) \left(\frac{1}{D_i \Delta x}\right) (T_I - T_{i,j}) \, .$$

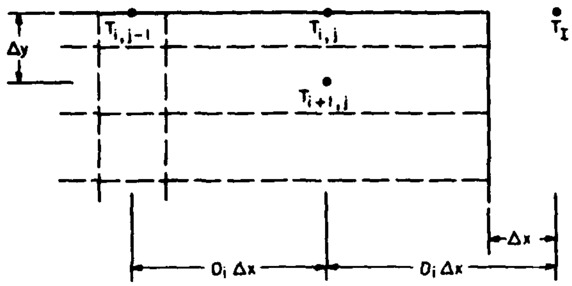



*Figure 7. Node on a corner of semi-infinite boundary.*

The total heat balance for steady state with $\Delta x = \Delta y = \Delta s$ is given by

$$\left(\frac{1}{\frac{1}{k_{i,j-1}} + \frac{2D_i-1}{k_{i,j}}}\right) T_{i,j-1} + \left(\frac{2(2D_i-1)}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}}\right) T_{i+1,j}$$

$$- \left(\frac{1}{\frac{1}{k_{i,j-1}} + \frac{2D_i-1}{k_{i,j}}} + \frac{2(2D_i-1)}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}} + \frac{1}{\frac{1}{k_I} + \frac{2D_i-1}{k_{i,j}}}\right) T_{i,j}$$

$$= -\phi\,(2D_i-1)\,\Delta s - \left(\frac{1}{\frac{1}{k_I} + \frac{2D_i-1}{k_{i,j}}}\right) T_I\,. \tag{21}$$

As is the case with the other semi-infinite boundaries, the node adjacent to the semi-infinite node must be given special consideration. Again, $\phi$ is the heat flux per unit area crossing the top of the grid.

$$Q_1 = \phi\Delta x,$$

$$Q_2 = \left(\frac{2}{\frac{1}{k_{i,j-1}} + \frac{1}{k_{i,j}}}\right)\left(\frac{\Delta y}{2}\right)\left(\frac{1}{\Delta x}\right)\left(T_{i,j-1} - T_{i,j}\right),$$

$$Q_3 = \left(\frac{2}{\frac{1}{k_{i+1,j}} + \frac{1}{k_{i,j}}}\right)\Delta x\left(\frac{1}{\Delta y}\right)\left(T_{i+1,j} - T_{i,j}\right),$$

$$Q_4 = \left(\frac{2D_i}{\frac{2D_i-1}{k_{i,j+1}} + \frac{1}{k_{i,j}}}\right)\left(\frac{\Delta y}{2}\right)\left(\frac{1}{D_i\Delta x}\right)\left(T_{i,j+1} - T_{i,j}\right).$$

Allow $\Delta x = \Delta y = \Delta s$. The heat balance equation for the node follows:

$$\left(\frac{1}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right) T_{i,j-1} + \left(\frac{2}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right) T_{i+1,j} + \left(\frac{1}{\frac{2D_i-1}{k_{i,j+1}}+\frac{1}{k_{i,j}}}\right) T_{i,j+1}$$

$$- \left(\frac{1}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}} + \frac{2}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}} + \frac{1}{\frac{2D_i-1}{k_{i,j+1}}+\frac{1}{k_{i,j}}}\right) T_{i,j} = -\phi\Delta s\,. \tag{22}$$

*Corner with convection on one side, semi-infinite on the other*

Again consider the corner illustrated in Figure 7, but allow convection to occur across the top of the grid. Let $h$ represent the convective heat transfer coefficient, and let $T_A$ be the ambient temperature outside the top of the grid; then the heat flows $Q_2$, $Q_3$, and $Q_4$ will be the same as those given for eq 21 and

$$Q_1 = h(2D_i-1)\,\Delta s\,(T_A - T_{i,j})\,.$$

The heat balance for the corner is given by

$$\left(\frac{1}{\frac{1}{k_{i,j-1}}+\frac{2D_i-1}{k_{i,j}}}\right) T_{i,j-1} + \left(\frac{2(2D_i-1)}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right) T_{i+1,j} - \left(\frac{1}{\frac{1}{k_{i,j-1}}+\frac{2D_i-1}{k_{i,j}}}\right.$$

$$\left. + \frac{2(2D_i-1)}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}} + \frac{1}{\frac{1}{k_I}+\frac{2D_i-1}{k_{i,j}}} + h(2D_i-1)\,\Delta s\right) T_{i,j} = -h(2D_i-1)\,\Delta s\,T_A$$

$$- \left(\frac{1}{\frac{1}{k_I}+\frac{2D_i-1}{k_{i,j}}}\right) T_I\,. \tag{23}$$

For the node adjacent to the corner node, illustrated in Figure 8, $Q_2$, $Q_3$, and $Q_4$ will be the same as those given for eq 22, and

$$Q_1 = h\Delta s(T_A - T_{i,j})\,.$$

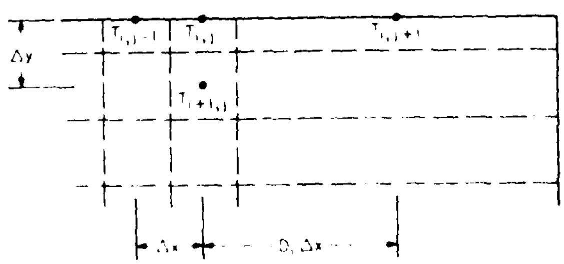



*Figure 8. Node adjacent to semi-infinite corner node.*

The heat balance for that node is given by

$$\left(\frac{1}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}}\right) T_{i,j-1} + \left(\frac{2}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}}\right) T_{i+1,j} + \left(\frac{1}{\frac{2D_i-1}{k_{i,j+1}}+\frac{1}{k_{i,j}}}\right) T_{i,j+1}$$

$$-\left(\frac{1}{\frac{1}{k_{i,j-1}}+\frac{1}{k_{i,j}}} + \frac{2}{\frac{1}{k_{i+1,j}}+\frac{1}{k_{i,j}}} + \frac{1}{\frac{2D_i-1}{k_{i,j+1}}+\frac{1}{k_{i,j}}} + h\Delta s\right) T_{i,j} = -h\Delta s T_A \, . \quad (24)$$

**Computer program development**

In order to solve a particular problem, a grid whose boundaries coincide with those in the problem must be set up. The number of grid points used is largely a matter of trial and error; as previously stated, the smaller the $\Delta s$ between the nodes (and hence the more dense the grid), the more accurate the solution. Once the grid is set up, each node in it is assigned the finite difference equation which represents the heat balance between that node and its adjacent nodes. The resulting system of equations may then be solved simultaneously to arrive at the steady-state temperature distribution within the grid.

SSCONDUCT, the program developed to model two-dimensional steady-state heat conduction, is made up of four parts: 1) a data gathering subroutine, 2) the main program, 3) a subroutine to solve the system of equations, and 4) a subroutine to locate the isotherms in the resulting temperature distribution. The grid for the problem is represented by a three-dimensional array, RAY (I, J, K). The first two dimensions designate the spatial location in Cartesian coordinates. The third dimension, K, has three values. RAY (I, J, 1) contains the temperatures for each nodal location I, J. RAY (I, J, 2) describes the location type of each node. Examples of location types include a node on a zero flux boundary, a constant temperature node, a variable interior node, etc. RAY (I, J, 3) is an index of the material type of each node. This index is used to assign conductivities and convection coefficients. It is possible to have a different material at each point in the grid.

*Data subroutine*

Subroutine SSDATA was written to provide a quick and easy way for the user to set up the grid, without having to worry about formats in the data file. The user has only to edit SSDATA, following directions in the comment statements in the computer program, to insert the desired values of the variables. When SSDATA is run, the new values of the variables are put into the formatted data file STSDAT. The main program then reads the data from STSDAT.

*Main program*

In the main part of SSCONDUCT, each node of the grid is examined. The conductivities are figured between the node being examined and the adjacent nodes, and the coefficients for the node's difference equation are figured and stored. A grid whose dimensions are X by Y contains XY nodes; hence there will by XY equations with XY unknowns. Since the matrix of coefficients for this system of equations is banded, with all entries being zero outside the band, only the entries of the band need to be stored. The bandwidth is $2X+1$. Thus, instead of storing an $XY$ by $XY$ array for the matrix of coefficients, an XY by 2X+1 array is stored. After each node in the grid has been examined, subroutine BANDMX is called to solve the system of equations.

*Subroutine BANDMX*

The International Mathematical and Statistical Library's subroutine LEQT1B has been adopted for solving the system of equations which has been stored in band form. Please consult Martin and Williamson (1967) for a detailed discussion of the method.

*Subroutine ISOTHM*

This subroutine examines the final temperature distribution in the grid, performing a linear interpolation between grid point temperatures to determine the spatial coordinates of user-specified isotherms. The coordinates are written into file POINTS which may then be plotted.

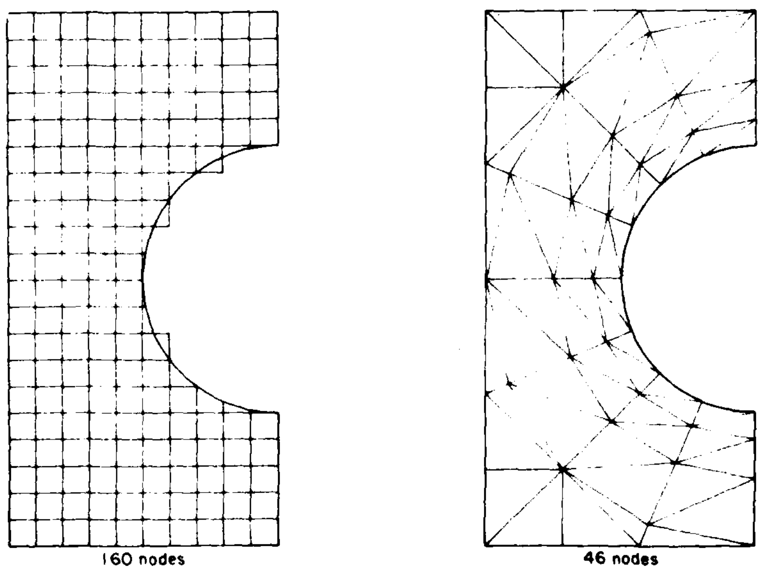
## FINITE ELEMENT METHODS

### Introduction

Finite element methods are a relatively recent addition to numerical methods. They can be used to obtain approximate solutions to governing differential equations encountered in many disciplines. They were first used by the aerospace industry during the 1950's for structural analysis of complex systems. By the 1960's people found that the method could be applied to a broad class of problems.

Finite element models use a piecewise approximation to the governing differential equations over the region of interest; finite difference models use a pointwise approximation. A major advantage of the finite element approximation over a finite difference approximation is its ability to model irregularly shaped boundaries more accurately. The size of the elements can also be easily varied. The basic shape may also be varied. Figure 9 shows how an irregularly shaped boundary would be modeled with both the finite element and finite difference methods. Each of the models approximates the circular boundary; however, the finite difference model uses a rather crude approximation in comparison. Another major advantage of the finite element method is apparent from Figure 9, that is, we can easily vary the element size. The geometry here is a half symmetry of a buried pipe. We expect the temperature gradients to be much greater in areas nearer the pipe, and by varying element size these areas can be modeled to any degree of accuracy, while surrounding areas with smaller temperature gradients need not be.

The finite element method does have several disadvantages. Since the grid is usually oddly shaped, defining all the coordinates of each node can be a tedious job. Automatic-mesh-generation




*a. Finite difference model.*　　*b. Finite element model.*

*Figure 9. Finite difference and finite element grids for a circular boundary.*

computer programs are available but not necessarily easy to use or inexpensive. Another disadvantage of the finite element method is its complexity. The necessary equations cannot always be entirely based on physical considerations, as can finite difference equations. Finite element computer programs tend to be complex and very hard to modify. In spite of these disadvantages, the finite element method is widely used in many disciplines of engineering today.

Before going into the development of a finite element model for heat conduction, we will first consider the basic steps which must be followed when using the finite element method to solve any problem.

The first step is to discretize the continuum, that is, divide the region to be modeled into elements. Many different shapes can be used for the element. Here we will consider only the most simple two-dimensional element, the triangle. The nodes and elements must now be numbered. As we will see later, this must be done carefully in order to get the most economical solution.

The next step is to select the type of interpolating function which will be used over the element to represent the field variable. Normally, polynomials are chosen since they are easily differentiated and integrated. Since we will be using a simple triangle with only three points our interpolation functions will be linear.

Next we need to find the equations which express the properties of each element. These will be in matrix form. Several approaches are possible. We will rely on the variational principal, which is known for the heat conduction problem. This approach is most convenient, but in some cases a variational principal does not exist for the problem. In these cases, either the direct approach, the energy balance method, or the method of weighted residuals (Galerkin's appraoch) may be used (Huebner 1975).

Now the element equations can be assembled into the global equations describing the entire region. This is a straightforward procedure easily handled by the computer. The system of equations must be modified to account for any boundary conditions present. The result is a system of simultaneous equations.

The next step is to solve this system of equations. This is usually the most time consuming step (computer time). With linear equations, as we will have, many methods exist for solution. The solution is somewhat simplified because the resulting matrix is banded and symmetrical. It is also positive definite. Round-off errors are also reduced because of these properties.

Once we have the solution we may want to compute additional quantities, such as isotherms or temperature gradients in the case of heat conduction.

In the next section we will discuss the procedure used at each of these steps by developing a two-dimensional finite element model for heat conduction. Various types of boundary conditions will be included in this model.

**Application to two-dimensional steady-state heat transfer**

In this section we will develop a finite element model for heat conduction in two dimensions. The simplest two-dimensional element, the triangle, will be used exclusively. Linear interpolation functions will be used within the elements and provisions for internal heat generation will be included. In an attempt to make the method as clear as possible we will follow the sequence of steps outlined in the last section.

The first step is to divide the region to be modeled into triangular elements. During this step we should try to concentrate the smaller elements in the areas of the highest gradients. Larger elements can be used in areas of smaller gradients. The triangles should be as close to equilateral as possible in order to promote accuracy of the solution. As an example, consider a problem with a geometry similar to that of Figure 9. The region can be divided into triangular elements in a similar fashion as shown in Figure 10.

Now the nodes need to be numbered. Care must be taken here in order to reduce computational requirements. As stated before, the simultaneous equations which result in a finite element

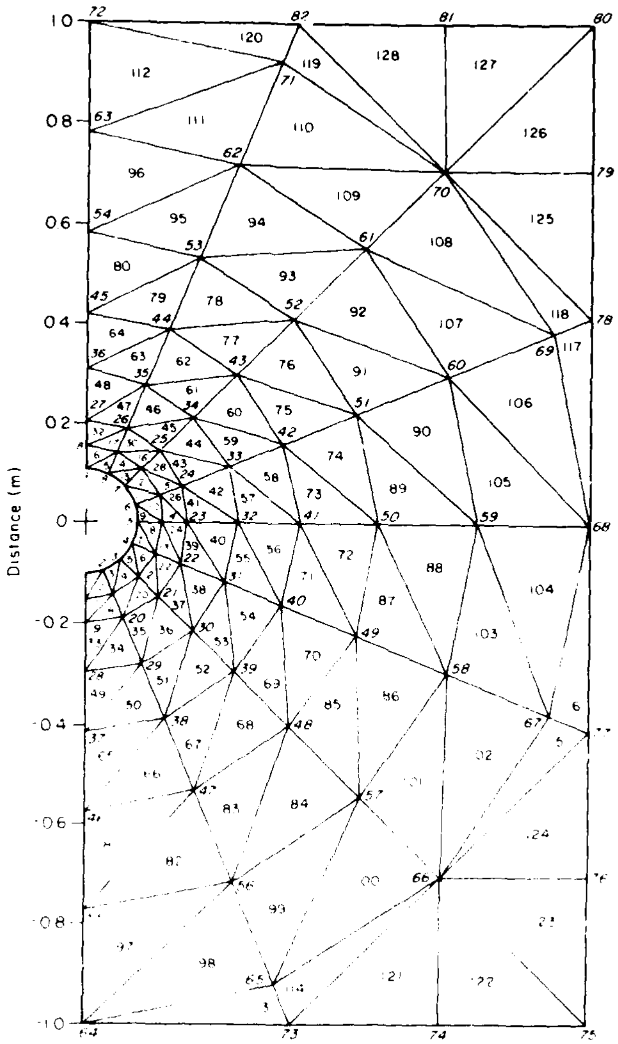


*Figure 10. A finite element grid.*

equation are banded when represented in matrix form. The smaller the bandwidth, the easier the equations are to solve. Storage requirements are also reduced since only the band need be stored. The numbering of the nodes directly affects the bandwidth of the matrix. The bandwidth is equal to one plus the largest difference between node numbers for any single element (out of all the elements) for a problem with one degree of freedom (temperature) at each node. In Figure 10 the nodes have been numbered in a way that we hope will minimize the bandwidth.

The elements must also be numbered. They may be numbered in any sequence, however, without affecting the computational efficiency of the solution. The computational procedure will require that we know which node numbers are associated with each element. We must also be able to identify what material a particular elements is in, if more than one material exists in the problem. A table in the format of Table 1 will accomplish this purpose.

**Table 1. Typical element data.**

| Element number | Nodes 1 | 2 | 3 | Material type |
|---|---|---|---|---|
| 1 | 10 | 2 | 1 | 1 |
| 2 | 10 | 11 | 2 | 1 |
| 3 | 11 | 3 | 2 | 1 |
| 4 | 11 | 12 | 3 | 1 |
| 5 | 12 | 4 | 3 | 1 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |

The local node numbers (1, 2, 3 in Table 1) for each element should always be consistently numbered with a particular convention. In this case they are numbered counterclockwise around the element.

We also need to define the spatial coordinates of each of the nodal points. A simple table like Table 2 will accomplish this.

**Table 2. Typical nodal point data.**

| Node | X | Y |
|---|---|---|
| 1 | 0.0000 | -0.3280 |
| 2 | 0.1256 | -0.3031 |
| 3 | 0.2319 | -0.2319 |
| 4 | 0.3031 | -0.1286 |
| 5 | 0.3280 | 0.0000 |
| . | . | . |
| . | . | . |
| . | . | . |

The coordinates given are for the nodes in Figure 10. The origin in this case is located at the center of the pipe.

Now we need to define the interpolation functions for the elements. Certain continuity requirements must be met by the element type and the interpolation functions, depending on the type of problem being studied. It is beyond the scope of this report to develop these requirements. In this case, the requirements are met by triangular elements and the linear interpolation functions which will be used.

The interpolation function must define the field variable, in this case temperature, over the entire element. That is, given the position of any point within the element, the element interpolation function gives an estimate of the temperature at that point. The form of the linear interpolating polynomial is

$$T = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{25}$$

where $T$ is the temperature, $x$ and $y$ are the coordinates and $\alpha_1$, $\alpha_2$ and $\alpha_3$ are constants.

At local nodes 1, 2 and 3 of the element the following conditions must be satisfied by the interpolating polynomial:

$T = T_1$ at $x = X_1$ and $y = Y_1$

$T = T_2$ at $x = X_2$ and $y = Y_2$

$$T = T_3 \text{ at } x = X_3 \quad \text{and} \quad y = Y_3 . \tag{26}$$

Using each of these conditions and solving for $T$ at each node gives us the following set of equations:

$$T_1 = \alpha_1 + \alpha_2 X_1 + \alpha_3 Y_1$$

$$T_2 = \alpha_1 + \alpha_2 X_2 + \alpha_3 Y_2$$

$$T_3 = \alpha_1 + \alpha_2 X_3 + \alpha_3 Y_3 . \tag{27}$$

Now we can solve this set of equations for *the constants* $\alpha_1, \alpha_2, \alpha_3$, yielding

$$\alpha_1 = \frac{1}{2A} \left[ (X_2 Y_3 - X_3 Y_2) T_1 + (X_3 Y_1 - X_1 Y_3) T_2 + (X_1 Y_2 - X_2 Y_1) T_3 \right]$$

$$\alpha_2 = \frac{1}{2A} \left[ (Y_2 Y_3) T_1 + (Y_3 Y_1) T_2 + (Y_1 - Y_2) T_3 \right]$$

$$\alpha_3 = \frac{1}{2A} \left[ (X_3 X_2) T_1 + (X_1 X_3) T_2 + (X_2 - X_1) T_3 \right] . \tag{28}$$

where $A$ is the area of the triangular element and $2A$ is given by the determinant of the following $3 \times 3$ matrix:

$$2A = \begin{vmatrix} 1 & X_1 & Y_1 \\ 1 & X_2 & Y_2 \\ 1 & X_3 & Y_3 \end{vmatrix} \tag{29}$$

Now we need to find the interpolation function in terms of the temperatures at each node and *the element topography*. We can do this by substituting our expression for the constants $\alpha_1$, $\alpha_2$, $\alpha_3$ into the original interpolation function (eq 25). The result will be an equation for $T$

$$T^{(e)} = N_1 T_1 + N_2 T_2 + N_3 T_3$$

or

$$T^{(e)} = [N] \{T\}^{(e)} . \tag{30}$$

Here the superscript (e) means "within the element" and $N_1$, $N_2$ and $N_3$ are the shape functions; they are given by the following equations:

$$N_1 = \frac{1}{2A} [a_1 + b_1 x + c_1 y]$$

$$N_2 = \frac{1}{2A} [a_2 + b_2 x + c_2 y]$$

$$N_3 = \frac{1}{2A} [a_3 + b_3 x + c_3 y] . \tag{31}$$

The constants in these equations are given by

$$a_1 = X_2 Y_3 - X_3 Y_2$$

$$a_2 = X_3 Y_1 - X_1 Y_3$$

$$a_3 = X_1 Y_2 - X_2 Y_1$$

$$b_1 = Y_2 - Y_3$$
$$b_2 = Y_3 - Y_1$$
$$b_3 = Y_1 - Y_2$$
$$c_1 = X_3 - X_2$$
$$c_2 = X_1 - X_3$$
$$c_3 = X_2 - X_1 . \tag{32}$$

It is easily shown by substituting back into eq 31 that, for any particular node, the corresponding shape function takes on a value of unity at that node and zero at the other two nodes and the line connecting them.

Now we can use the information in Tables 1 and 2 to assemble the individual element equations into a global system of equations. The result is a set of piecewise continuous equations to approximate temperature over the entire region. Our next step is to relate this to the physical problem of heat conduction in a solid medium.

Several methods are available for finding the element equations, as mentioned earlier. For this problem we will use the variational principal, which is known. A discussion of the calculus of variations is beyond the scope of this report. The interested reader is referred to Pars (1962), Schechter (1967), Huebner (1975), Segerlind (1976) and Zienkiewicz (1977).

The calculus of variations simply states that, if we can find the so-called "functional" which corresponds to the governing differential equation and boundary conditions of the problem, the minimization of that functional requires that its corresponding differential equation and boundary conditions be satisfied. For isotropic two-dimensional steady-state heat transfer, the governing differential equation is

$$\frac{\partial}{\partial x}\left(k\,\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\,\frac{\partial T}{\partial y}\right) + Q_I = 0 \tag{33}$$

where $Q_I$ is the internal heat generation within the region. Constant temperature boundary conditions are simply

$$T = T_B \quad \text{on} \quad S_1 \tag{34}$$

where

$T_B$ = constant temperature of the boundary $S_1$
$S_1$ = segment of boundary at $T_B$.

Constant heat flux boundary conditions are given by

$$k\,\frac{\partial T}{\partial x}\,\ell_x + k\,\frac{\partial T}{\partial y}\,\ell_y - \phi = 0 \quad \text{on} \quad S_2 \tag{35}$$

where

$\ell_x$ and $\ell_y$ = direction cosines of a vector perpendicular to $S_2$.
$S_2$ = surface subject to the constant heat flux $\phi$.

Finally, the conditions on a convective boundary are

$$k\,\frac{\partial T}{\partial x}\,\ell_x + k\,\frac{\partial T}{\partial y}\,\ell_y + h(T - T_A) = 0 \quad \text{on} \quad S_3 \tag{36}$$

where $S_3$ is the boundary segment subject to convection.

The entire boundary of the region is made up completely of segments $S_1, S_2$ and $S_3$. Radiative boundary conditions are not considered here.

Now we need to use the variational principal to find the element equations. The functional matching eq 33 with boundary conditions (eq 34, 35 and 36) is (Schechter 1967, Huebner 1975, Segerlind 1976)

$$I(T) = \frac{1}{2} \int_A \left[ k\left(\frac{\partial T}{\partial x}\right)^2 + k\left(\frac{\partial T}{\partial y}\right)^2 - 2Q_1 T \right] dx\, dy + \int_{S_4} \left[\phi T + 1/2h(T-T_A)^2\right] dS_4 \qquad (37)$$

where $S_4$ is the union of surfaces $S_2$ and $S_3$.

We can define $T$ over the element by recalling eq 30

$$T^{(e)} = \lfloor N \rfloor \{T\}^{(e)} .$$

To minimize the functional $I(T)$, which will in turn satisfy governing differential equation and boundary conditions, we can set its derivative with respect to temperature $T$ equal to zero. Because our interpolation functions meet the continuity requirements as discussed earlier, the total integral $I(T)$ is the sum of the integral $I(T^{(e)})$ for all the elements. If we have $M$ elements,

$$I(T) = \sum_{e=1}^{M} I(T^{(e)}) . \qquad (38)$$

Because the temperatures at each node of an element are independent, the partial derivative of the integral $I(T^{(e)})$ with respect to each nodal temperature must be zero:

$$\frac{\partial I(T^{(e)})}{\partial T_1} = \frac{\partial I(T^{(e)})}{\partial T_2} = \frac{\partial I(T^{(e)})}{\partial T_3} = 0 . \qquad (39)$$

If node 1 is on the boundary of surface $S_4$, we have

$$\frac{\partial I(T^{(e)})}{\partial T_1} = 0 = \int_{A^{(e)}} \left[ k \frac{\partial T^{(e)}}{\partial x} \frac{\partial}{\partial T_1}\left(\frac{\partial T^{(e)}}{\partial x}\right) + k \frac{\partial T^{(e)}}{\partial y} \frac{\partial}{\partial T_1}\left(\frac{\partial T^{(e)}}{\partial y}\right) \right.$$

$$\left. + Q_1 \frac{\partial T^{(e)}}{\partial T_1} \right] dA^{(e)} + \int_{S_4^{(e)}} \left(\phi \frac{\partial T^{(e)}}{\partial T_1} + h(T-T_A)^{(e)} \frac{\partial T^{(e)}}{\partial T_1}\right) dS_4 . \qquad (40)$$

Similar equations apply to other nodes on boundary $S_4$. For nodes not on boundary $S_4$, the integral over $S_4$ is dropped. The derivatives in the above equation can be evaluated by considering eq 30. After substituting these into eq 40, we find the result for node 1 on boundary $S_4$ is

$$\frac{\partial I(T^{(e)})}{\partial T_1} = 0 = \int_{A^{(e)}} \left[ k \left\lfloor \frac{\partial N}{\partial x} \right\rfloor \{T\} \frac{\partial N_1}{\partial x} + k \left\lfloor \frac{\partial N}{\partial y} \right\rfloor \{T\} \frac{\partial N_1}{\partial y} + Q_1 N_1 \right] dA^{(e)}$$

$$+ \int_{S_4} \left[ \phi N_1 + h \lfloor N \rfloor N_1 \{T - T_A\} \right] dS_4^{(e)} . \qquad (41)$$

The element equation for each element is now found by combining the resulting three nodal equations,

$$\left\{\frac{\partial I}{\partial T^{(e)}}\right\}^{(e)} = \left[\frac{\partial I(T^{(e)})}{\partial T_1}, \frac{\partial I(T^{(e)})}{\partial T_2}, \frac{\partial I(T^{(e)})}{\partial T_3}\right] \quad \text{transpose}$$

$$= [K]^{(e)} \{T\}^{(e)} + \{R\}^{(e)} + [K_{S_4}]^{(e)} \{T\}^{(e)}$$

$$= \left[[K]^{(e)} + [K_{S_4}]^{(e)}\right] \{T\}^{(e)} + \{R\}^{(e)} = 0 . \tag{42}$$

Both the $[K]^{(e)}$ and $[K_{S_4}]^{(e)}$ are 3x3 matrices while $\{R\}^{(e)}$ and $\{T\}^{(e)}$ are 3x1 vectors. Typical terms within them are

$$K_{12} = \int_{A^{(e)}} k\left(\frac{\partial N_1}{\partial x}\frac{\partial N_2}{\partial x} + \frac{\partial N_1}{\partial y}\frac{\partial N_2}{\partial y}\right) dA^{(e)}$$

$$R_2 = \int_{A^{(e)}} Q_1 N_2 dA^{(e)} + \int_{S_4^{(e)}} \phi N_2 dS_4^{(e)}$$

$$K_{S_{4_{12}}} = \int_{S_4^{(e)}} h N_1 N_2 dS_4^{(e)} . \tag{43}$$

Now each of these equations for an element can be assembled into the global system of equations. The procedure for doing so is straightforward. If we have $n$ nodes in our problem, our resulting matrix will have dimensions of $n \times n$. Using the data in Table 1 we simply add all of the entries of each of the element matrices into the corresponding global position in the $n \times n$ system matrix. The same basic procedure applies to finding the global $R$ vector.

The assembled system of equations needs to be modified to account for constant temperature boundary conditions. Although exact methods are available for this procedure, we will use an approximate technique which is much simpler to program for computer execution. Before modification, eq 42 is rewritten into the form for solution

$$[K]\{T\} = \{F\} \tag{44}$$

where $\{K\}$ is now the sum of the global $[K]$ and $[K_{S_4}]$ matrices and $\{F\}$ is simply $-\{R\}$. If we have a constant temperature boundary condition at a given node we simply multiply the corresponding diagonal term in the $[K]$ matrix by a relatively large number and also replace the corresponding term in the $\{F\}$ vector by the product of the boundary temperature and this modified diagonal term. In this case we have used $10^{15}$ as the large number. This is an approximate procedure but it will yield good results as long as the boundary temperature specified is not very small (Segerlind 1976).

Now we are ready to solve for the unknown temperatures. If our problem has $n$ nodes our $[K]$ matrix will be an $n \times n$ matrix and both $\{T\}$ and $\{F\}$ will be $n \times 1$ vectors. It becomes obvious that for large problems, often having hundreds of nodes, this is the most time consuming step. For this reason it is imperative that our solution method be efficient. The fact that our resulting $[K]$ matrix has some special characteristics, as mentioned earlier, greatly simplifies and speeds the solution.

Many techniques are available for solving linear sets of equations such as we have here. The interested reader may refer to one of the many texts on the subject, including Forsythe et al.

(1977) and Gerald (1978). The method used here is in the form of a computer program subroutine. The subroutine is called MCHB and is part of the IBM SSP (Scientific Subroutine Package), which is available on many computer systems. The matrix which is provided to MCHB must be in compressed form, consisting of only the main diagonal and the upper portion of the band, which is symmetrical. The terms are stored in rows in successive storage locations. It returns the temperature, after solution, in the same space that the $\{F\}$ vector was supplied in. The Cholesky decomposition technique is used by MCHB to solve the matrix equations.

All the steps of solution for a two-dimensional steady-state heat transfer problem using the finite element method have now been outlined. The only remaining step is to compute additional quantities based on the resulting temperature distribution. Often, the location of a particular isotherm may be of interest. A simply method of finding the approximate location of an isotherm is simply to interpolate linearly along the element boundary between nodes whose temperatures bound the isotherm of interest.

In the next section we will discuss the computer program based on the theory developed in this section.

**Finite element computer program**

A finite element computer program, called FEHEAT, has been developed to model two-dimensional steady-state heat conduction. This program is based in part on a program presented by Huebner (1975). The following types of boundary conditions have been included:

1. Constant temperature
2. Specified heat flux (including, of course, zero flux insulated boundaries)
3. Convective heat transfer.

FEHEAT also has the capability to account for internal heat generation.

The program consists of a main program and five subroutines. The main program is responsible for the following operations:

1. Providing dimensioned and common storage space for the necessary matrices, vectors, and scalars
2. Initializing all of the above to zero, as required
3. Determining the size of the problem and type of boundary conditions present
4. Reading in all input data and printing them out for confirmation.
5. Determining the bandwidth of the $[K]$ matrix
6. Calling the appropriate subroutine for solution and isotherm location
7. Printing out resulting temperatures' and isotherms' coordinates

The names and purposes of the five subroutines are:

1. TSM This subroutine forms the $[K]^{(e)}$ matrix (element "thermal stiffness" matrix). A different type of material may be used in each element. TSM computes the constants in the linear interpolation functions as well as the element area. The "thermal stiffness" matrix is modified to account for convection and the internal heat generation at each node is computed here for use in subroutine FRHS.

2. FRHS This subroutine forms the $\{F\}$ vector (right hand side). It begins with the contribution of internal heat generation at the affected nodes and adds to that the effect of specified heat fluxes and convective boundary segments. It is then modified for the constant temperature boundary conditions by the procedures outlined in the previous section.

3. FORMK This subroutine assembles the element $[K]^{(e)}$ matrices to form the global $[K]$ matrix. Constant temperature boundary conditions are accounted for by modification of the corresponding diagonal element as outlined earlier. The resulting matrix is stored, in rows first then in columns, in consecutive storage locations, with only the diagonal and upper portion of the band being stored, as required by MCHB. This procedure greatly reduces storage requirements for banded matrices.

4. MCHB—This subroutine solves the system of equations. The $[K]$ matrix must be stored in the form explained earlier. Double precision is used on several crucial quantities. This subroutine is part of the IBM Scientific Subroutine Package available on many computers.

5. ISOTHM—This subroutine finds the location of any number of specified isotherms at any temperature. It examines each element and performs linear interpolation between adjacent nodes within an element which bounds the temperature of interest. Its results are sets of coordinates for each temperature specified by the user. These results can be plotted with a simple plotting program.

More will be said on the computer program and the preparation of input data in the next section where we examine its application and results for a classical problem. A listing of the computer program, including all the subroutines and sample input files and output data, is given in Appendix B.

## PROGRAM VERIFICATIONS AND COMPARISONS

Each computer program was run modeling two steady-state two-dimensional problems: the case of a rectangular plate subject to a uniform temperature on three sides and a sinusoidal temperature distribution across the fourth side, and the case of a buried pipe. Analytical solutions exist for both of these problems. In the *Rectangular plate* section, the rectangular plate problem is solved analytically, and the results are compared to the results of each program. In the *Buried pipe* section, the analytical solution of the buried pipe is presented, along with comparisons of the computer-generated solutions. In the *Semi-infinite condition* section, a one-dimensional problem is solved by the finite difference program to illustrate the use of the semi-infinite boundary.

### Rectangular plate

Consider a rectangular plate that has three of its sides at a fixed temperature and has a sinusoidal temperature distribution across the fourth side. The following well-known analytical solution exists to find the steady-state temperature distribution within the plate.

Assume the plate is sufficiently thick so that the end effects may be neglected. The thermal conductivity is constant throughout the plate. We examine a cross section of the plate, and use a shifted temperature $\theta = T - T_0$. The heat flow obeys Laplace's equation

$$\frac{\partial^2 \theta}{\partial x^2} + \frac{\partial^2 \theta}{\partial y^2} = 0 .$$

The boundary condition on the top surface is $\theta = \theta_m \sin \pi x/W$; $\theta = 0$ for the three other surfaces. This gives us the following boundary conditions:

(1) $\theta\ (0, y) = 0$

(2) $\theta\ (W, y) = 0$

(3) $\theta\ (x, 0) = 0$

(4) $\theta\ (x, H) = \theta_m \sin \frac{\pi x}{W}$ . (45)

We begin by assuming a solution of the form

$$\theta(x, y) = X(x)Y(y) .$$

Then, from Laplace's equation

$$-\frac{1}{X}\frac{\partial^2 X}{\partial x^2} = \frac{1}{Y}\frac{\partial^2 Y}{\partial y^2} .$$

We then proceed as usual with the method of separation of variables, where the separation constant is $\lambda^2$:

$$-\frac{1}{X}\frac{\partial^2 X}{\partial x^2} = \lambda^2, \quad \frac{1}{Y}\frac{\partial^2 Y}{\partial y^2} = \lambda^2$$

$$\frac{d^2 X}{dx^2} + \lambda^2 X = 0, \quad \frac{d^2 Y}{dy^2} - \lambda^2 Y = 0$$

$$X = C_1 \cos \lambda x + C_2 \sin \lambda x, \quad Y = C_3 e^{-\lambda y} + C_4 e^{\lambda y}$$

$$\theta = (C_1 \cos \lambda x + C_2 \sin \lambda x)(C_3 e^{-\lambda y} + C_4 e^{\lambda y}) .$$

Apply the boundary conditions, from eq $45_1$,

$$C_1 (C_3 e^{-\lambda y} + C_4 e^{\lambda y}) = 0$$

$$C_1 = 0 .$$

From eq $45_2$,

$$(C_2 \cos \lambda x + C_2 \sin \lambda x)(C_3 + C_4) = 0$$

$$C_3 = -C_4 .$$

From eq $45_3$, with results of eq $45_1$ and $45_2$,

$$C_2 C_4 \sin \lambda W (e^{\lambda y} - e^{-\lambda y}) = 0$$

$$2 C_2 C_4 \sin \lambda W \sinh \lambda y = 0 .$$

This requires that $\sin \lambda W = 0$ or $\lambda = n\pi/W$ ($n$ is an integer). Because of the linearity of Laplace's equation, $\theta$ can be written as a sum of an infinite series

$$\theta = \sum_{n=1}^{\infty} C_n \sin \frac{n\pi x}{W} \sinh \frac{n\pi y}{W}$$

where the constants have been combined.

Now apply the boundary condition from eq $45_4$:

$$\theta_m \sin \frac{\pi x}{W} = \sum_{n=1}^{\infty} C_n \sin \frac{n\pi x}{W} \sinh \frac{n\pi H}{W} .$$

This holds only if

$$C_1 = \frac{\theta_m}{\sinh \frac{\pi H}{W}}$$

and if $C_2 = C_3 = \ldots = C_n = 0$, then the final temperature distribution in the plate is given by

$$\theta = \theta_m \frac{\sinh \frac{\pi y}{W}}{\sinh \frac{\pi H}{W'}} \sin \frac{\pi x}{W}$$

or,

$$T = T_o + T_m \frac{\sinh \frac{\pi y}{W}}{\sinh \frac{\pi H}{W'}} \sin \frac{\pi x}{W} \quad . \tag{46}$$

For the computer comparisons, let $T_o = 100°C$, $T_m = 50°C$, $H = 8$ m, and $W' = 12$ m. The results were calculated for every 1 m increment of space in the rectangle, and the results are shown in Table 3.

**Table 3. The rectangular plate solution.**

| | 0* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 100.00† | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| 1 | 100.00 | 100.86 | 101.66 | 102.34 | 102.87 | 103.20 | 103.31 | 103.20 | 102.87 | 102.34 | 101.66 | 100.86 | 100.00 |
| 2 | 100.00 | 101.77 | 103.43 | 104.84 | 105.93 | 106.62 | 106.85 | 106.62 | 105.93 | 104.84 | 103.43 | 101.77 | 100.00 |
| 3 | 100.00 | 102.81 | 105.43 | 107.68 | 109.41 | 110.49 | 110.86 | 110.49 | 109.41 | 107.68 | 105.43 | 102.81 | 100.00 |
| 4 | 100.00 | 104.04 | 107.81 | 111.05 | 113.53 | 115.09 | 115.62 | 115.09 | 113.53 | 111.05 | 107.81 | 104.04 | 100.00 |
| 5 | 100.00 | 105.55 | 110.73 | 115.17 | 118.58 | 120.73 | 121.46 | 120.73 | 118.58 | 115.17 | 110.73 | 105.55 | 100.00 |
| 6 | 100.00 | 107.45 | 114.39 | 120.35 | 124.92 | 127.80 | 128.78 | 127.80 | 124.92 | 120.35 | 114.39 | 107.45 | 100.00 |
| 7 | 100.00 | 109.85 | 119.04 | 126.92 | 132.97 | 136.78 | 138.08 | 136.78 | 132.97 | 126.92 | 119.04 | 109.85 | 100.90 |
| 8 | 100.00 | 112.94 | 125.00 | 135.36 | 143.30 | 148.30 | 150.00 | 148.30 | 143.30 | 135.36 | 125.00 | 112.94 | 100.00 |

*Increments in meters.
†Values in °C.

This problem was run on SSCONDUCT, the finite difference program, using a 17×25 grid (425 nodes) with an internodal spacing of 0.25 m. The results compare to within 0.01°C of the analytical solution.

FEHEAT, the finite element program, also used 425 nodes. The elements were isosceles right triangles with sides 0.25 m long. The total number of elements used was 768. The results from this model predicted the temperature to within 0.02°C of the analytical solution throughout the region.

### Buried pipe

The solution to the steady-state problem of a buried pipe may be found from vector field theory by superposition of the potentials of two line sources with equal and opposite strength.

In general, the gradient of a scalar field is a vector. In the case of a line source, we assume that the source is of uniform strength all along its length. Then the radial vectors are those which are of interest. In the case of heat flow, allow the radial vectors to be denoted by $\underline{q}$. The scalar potential field is the temperature distribution

$$\underline{q} = -k \underline{\nabla} T \tag{47}$$

where $k$ is the thermal conductivity, and $\underline{q}$ is given by $\underline{q} = W/2\pi r(\hat{\underline{r}})$, where $W$ is the source strength per unit length (W/m), $r$ is the distance from the line source to the point under consideration, and $\underline{r}$ is the unit vector in the radial direction pointing from the line source towards the point under consideration, as shown in Figure 11.

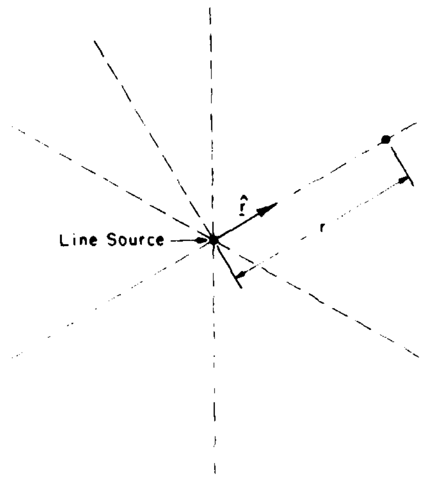



*Figure 11. Simple line source.*

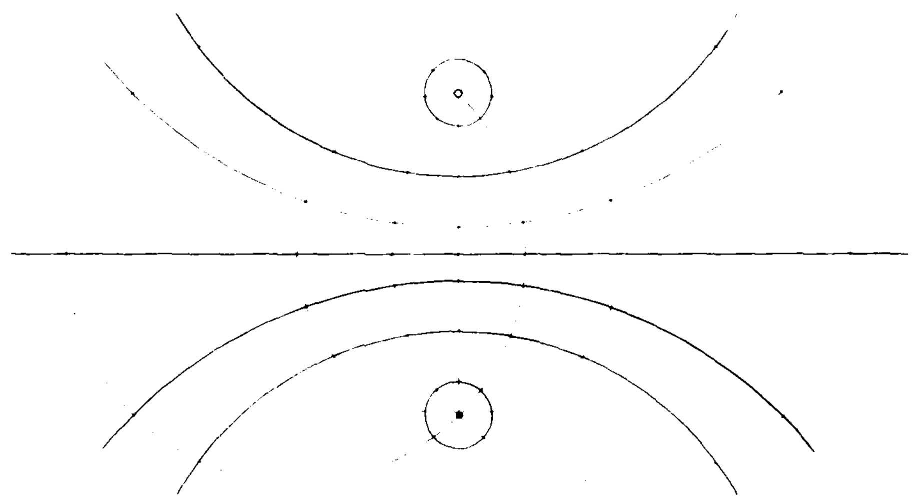


*Figure 12. Superposition of two line sources (●–heat source below the ground; ○–heat source above the ground).*

Upon substitution into eq 47:

$$\frac{W}{2\pi r}\,\hat{\underline{r}} = -k\,\frac{\partial T}{\partial r}\,\hat{\underline{r}}$$

$$\int -\frac{W}{2\pi k r}\,\hat{\underline{r}}\,dr = \int dT\,\hat{\underline{r}}$$

$$-\frac{W}{2\pi k}\,\ln\frac{r}{r_o} = T - T_o = \phi\,. \tag{48}$$

Now consider the situation illustrated in Figure 12. $W$ is the heat source strength below the ground; we also imagine a source of equal and opposite strength an equal distance above the ground. The field lines are the dotted lines, the isotherms are represented by continuous lines. For the bottom source

$$\phi_1 = -\frac{W}{2\pi k} \ln \frac{r_1}{r_o}$$

and, for the top source,

$$\phi_2 = \frac{W}{2\pi k} \ln \frac{r_2}{r_o} \, .$$

The resultant field (temperature distribution) may be obtained by superposing the potentials from these two fields,

$$\phi_1 + \phi_2 = \frac{W}{2\pi k} \left( \ln \frac{r_2}{r_o} - \ln \frac{r_1}{r_o} \right)$$

$$\phi = \frac{W'}{2\pi k} \ln \frac{r_2}{r_1}$$

$$T - T_o = \frac{W'}{2\pi k} \ln \frac{r_2}{r_1} \, . \tag{49}$$

Isotherms are lines of constant $\phi$. When $r_2 = r_1$, the straight line between the two fields is given by

$$\phi = \frac{W'}{2\pi k} \ln 1 = 0$$

$$T - T_o = 0$$

$$T = T_o \, .$$

Thus the reference temperature is given at the ground surface.

Now put the problem on a system of Cartesian coordinates, as illustrated in Figure 13.

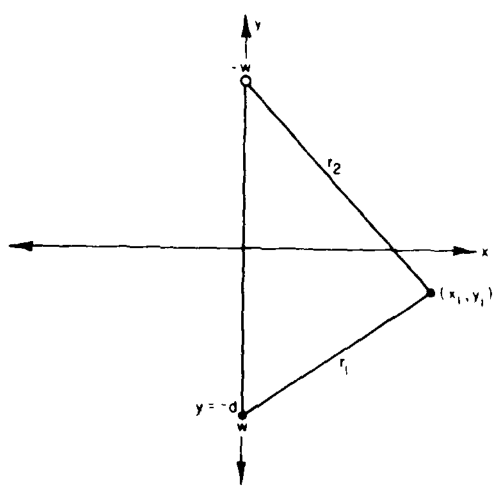



*Figure 13. Cartesian coordinates for source problem.*

The source of strength $W'$ is located at the point $(0, -d)$. From Figure 13, the distances $r_2$ and $r_1$ are given by

$$r_2 = \sqrt{x_1^2 + (d - y_1)^2}$$

$$r_1 = \sqrt{x_1^2 + (d+y_1)^2}\ .$$

Substitute these results into eq 49

$$T - T_o = \frac{W'}{2\pi k} \ln \frac{\sqrt{x_1^2 + (d-y_1)^2}}{\sqrt{x_1^2 + (d+y_1)^2}}$$

$$\frac{4\pi k(T - T_o)}{W'} = \ln \frac{x_1^2 + (d-y_1)^2}{x_1^2 + (d+y_1)^2}$$

$$\exp\left(\frac{4\pi k}{W'} (T - T_o)\right) = \frac{x_1^2 + (d-y_1)^2}{x_1^2 + (d+y_1)^2}\ . \tag{50}$$

This gives a general equation for the temperature at any point, given the ground surface temperature $T_o$, the conductivity, and the source strength. It will be shown below that the resulting isotherms have a circular shape. We will let the pipe be represented by one of the isotherms. Given the temperature of that isotherm, we may calculate the resulting temperature distribution, with the temperature of the pipe and the temperature of the ground surface also given.

For a given isotherm, the left-hand side of eq 50 is constant. Let

$$c = \exp\left(\frac{4\pi k}{W'} (T - T_o)\right),$$

then

$$c = \frac{x_1^2 + (d-y_1)^2}{x_1^2 + (d+y_1)^2}\ .$$

After the fraction is cleared in this equation, we complete the square to arrive at

$$x_1^2 + \left[y_1 + d\left(\frac{c+1}{c-1}\right)\right]^2 = d^2\left[\left(\frac{c+1}{c-1}\right)^2 - 1\right]\ .$$

Thus each isotherm is a circle with center $c$ and radius $r$, as given below:

$$c = (o, d)\left(\frac{c+l}{c-1}\right)$$

$$r = 2d\ \frac{\sqrt{c}}{c-1}\ .$$

For comparison with the computer models, consider the problem of a pipe with a 0.1-m radius at 100°C buried at a depth of 1 m (center of pipe); the ground surface is at 10°C.

The pipe is the 100°C isotherm whose center is at $1 = d\ (c_o + 1/c_o - 1)$

$$r = 0.1 = 2d\ \frac{\sqrt{c_o}}{c_o - 1}$$

Substitute for $d$ to obtain

$$0.1 = 2\left(\frac{c_o - 1}{c_o + 1}\right) \frac{\sqrt{c_o}}{(c_o - 1)}$$

$$c_o^2 - 398\, c_o + 1 = 0$$

$$c_o = 397.9975$$

$$d = \frac{c_o - 1}{c_o + 1} = 0.995\ .$$

Then

$$397.9975 = \exp[4\pi k/W(100-10)]$$

$$5.986 = 360\, \pi k/W$$

$$W = 360\, \pi k/5.986\ .$$

For any isotherm,

$$C = \exp\ [4\pi k(T - T_o/W)] = \exp[4\pi k(T - T_o)\ (5.986)/360\ \pi k]$$

$$C = \exp\ [0.0665\ (T - T_o)]\ .$$

Thus, given a point $(x, y)$ of interest, its temperature may be calculated from the following equation:

$$e^{0.0665(T-10)} = \frac{x_1^2 + (0.995 - y_1)^2}{x_1^2 + (0.995 + y_1)^2}$$

$$T = 10 + 15.03396\ \ln \frac{x_1^2 + (0.995 - y_1)^2}{x_1^2 + (0.995 + y_1)^2}\ . \qquad (51)$$

The region modeled by the computer programs was a rectangular area extending from the ground surface to 3 m below the surface, and 2 m to each side of the pipe. Because there is a vertical line of symmetry extending from the ground surface down through the center of the pipe, the computer program modeled only half of the area, and used a zero flux boundary along the right side, which passes through the center of the pipe.

The finite difference program used a 31x 21 grid (651 nodes), with an internodal distance of 0.1 meters. The pipe was represented by four constant temperature nodes. The left side and bottom boundaries were assigned the semi-infinite boundary condition. See Appendix A for the input and output for the program; the grid is labeled and printed in the output file, STDOUT. This problem required 88 CPU-seconds (Central Processing Unit) on CRREL's PRIME 400 computer, a cost of \$3.08.

The finite element program used 104 nodes (166 elements). The elements were triangles with sides of varying lengths, as shown in Figure 14. The left side and bottom boundaries were assigned a constant temperature boundary condition. The finite element problem required 26 CPU-seconds on the same computer, and cost \$0.91.

The three solutions–analytical, finite difference, and finite element are graphed in Figure 15 for the 30°, 50° and 70°C isotherms. Also shown are the finite difference and finite element 100°C nodes, which represent the pipe. Note that the finite element method is better able to

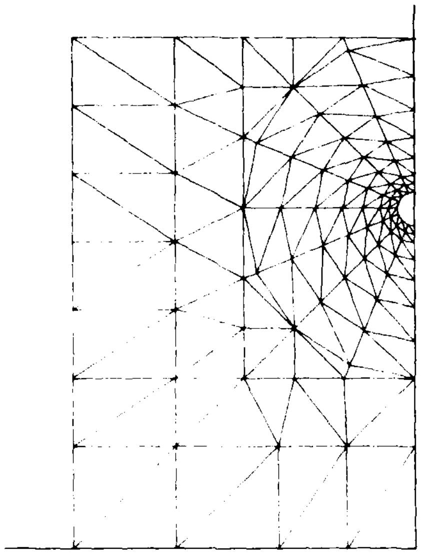


*Figure 14. Finite element grid for buried pipe problem.*

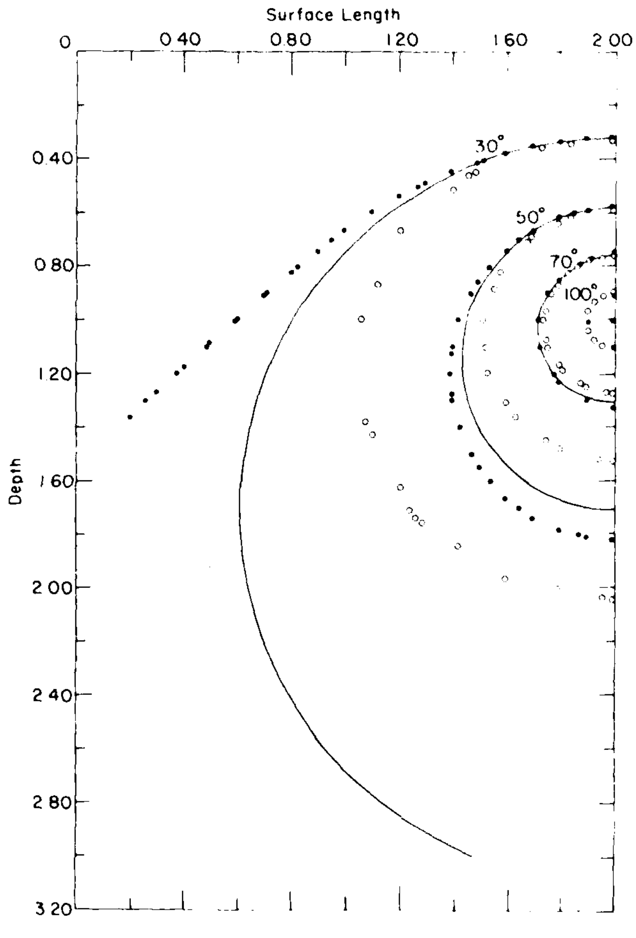



*Figure 15. Comparison of finite difference (●) and finite element ( ) results for buried pipe problem (the solid line is the analytical solution).*

model the circular surface of the pipe, while the finite difference method used only a four-node representation. Both methods represent the temperatures above the pipe very well, but the finite difference method is slightly more accurate. For regions to the left of the pipe and below the pipe, there are inaccuracies in both of the computed solutions, mainly due to the effect of the semi-infinite boundaries. The finite difference solution overpredicts the distance of the various isotherms from the pipe, but is the more accurate solution in this case. The finite element method underpredicts the isotherm locations. The internodal spacing used could be reduced in both cases should a more accurate solution be necessary.

**The semi-infinite condition**

The semi-infinite boundary condition in the finite difference program is modeled by use of a long rectangular node at the boundary. The temperature at infinity is known. To verify this approach, consider the one-dimensional problem of a semi-infinite plane. The temperature at the edge is 115°C, and the temperature at infinity is 0°C. If we allow "infinity" to be a large but finite distance, then a linear temperature distribution between 0° and 115°C would be expected at steady state.

A grid 17 nodes long was used to model this situation (Fig. 16). There are 16 nodes of regular shape in the grid, and one semi-infinite boundary node. Let *DS* be the internodal distance for the interior of the grid, and 100·*DS* be the distance from the last square node to the node at infinity. Listed in Table 4 are the expected temperatures for each node and those calculated by SSCONDUCT.

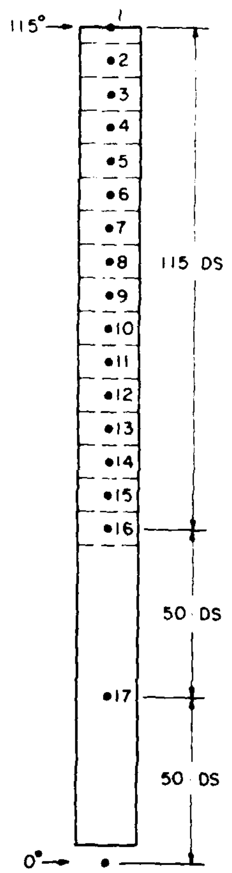



*Figure 16. Semi-infinite verification problem.*

**Table 4. One-dimensional semi-infinite comparison.**

| *Node* | *Temperature expected (°C)* | *Temperature calculated (°C)* |
|---|---|---|
| 1 | 115 | 115.00 |
| 2 | 114 | 114.00 |
| 3 | 113 | 113.00 |
| 4 | 112 | 111.99 |
| 5 | 111 | 110.99 |
| 6 | 110 | 109.99 |
| 7 | 109 | 108.99 |
| 8 | 108 | 107.98 |
| 9 | 107 | 106.98 |
| 10 | 106 | 105.98 |
| 11 | 105 | 104.98 |
| 12 | 104 | 103.97 |
| 13 | 103 | 102.97 |
| 14 | 102 | 101.97 |
| 15 | 101 | 100.97 |
| 16 | 100 | 99.97 |
| 17 | 50 | 49.88 |

The error at node 17 is small, considering that the distance modeled by the one rectangular node is almost seven times the size of the rest of the grid. Errors encountered when using the semi-infinite condition in two dimensions, however, may be much larger, as may be seen from the buried pipe example. In general the semi-infinite condition should be used in two-dimensional problems only in regions where the temperature gradient is small and precise knowledge of the temperature distribution is not critical.

## INSTRUCTIONS FOR USE OF COMPUTER PROGRAMS

This section is provided for those who wish to use either one of the two heat conduction programs. SSCONDUCT is the finite difference program, and FEHEAT is the finite element program.

### Instructions for SSCONDUCT

SSCONDUCT was set up to be easily modified to handle new problems. Running a new problem simply involves editing subroutine SSDATA by following instructions provided by comment statements in the program to adjust the variables and arrays. The variables are defined at the beginning of SSDATA.

However, before attempting to alter the data subroutine, the user should make a drawing of the problem. For a problem with more than one thermal conductivity, a number should be assigned to each different material in the problem, starting with 1. Let $A$ be the number of different materials in the problem. Assign the number $A$ to the material which appears the most in the problem. For each material, obtain the thermal conductivity in W/m K. If a material is located on a convective boundary, obtain the convection coefficient in W/m$^2$ K.

Now the finite difference grid for the problem should be drawn. The boundaries of the grid should match with those in the problem as closely as possible. Except for semi-infinite boundaries, the nodes should be equally spaced. At this point, a decision must be made on a reasonable nodal spacing. To date, there is no general way of determining the optimum nodal spacing, but there are the following guidelines:

1. The nodes should represent the materials in the grid as accurately as possible, i.e. no node should be composed partly of material 1 and partly of material 2.
2. Regions with a steep temperature gradient are best represented by a dense grid.
3. In general, accuracy decreases as internodal spacing increases.
4. The more nodes used, the more computer storage space is required, and the more computer time will be needed to solve the problem.

Bear in mind that the use of semi-infinite nodes introduces an error that is proportional to the area of the semi-infinite node. Therefore, the semi-infinite condition should be used only in regions of small temperature gradients when knowledge of the temperature distribution is not critical.

Let $y$ be the number of rows in the grid, and let $x$ equal the number of columns in the grid. The grid is represented in SSCONDUCT by RAY (I, J, K), where I is increased from 1 to $y$, J from 1 to $x$, and K from 1 to 3.

It may be desirable to make three drawings of the grid. The first time, label each node with temperatures where they are known. This will represent RAY (I, J, 1). The second time, label each node with its location type, as defined at the beginning of subroutine SSDATA. For example, if the top left-hand corner of the grid is at constant temperature, label it "11" (RAY [1, 1, 2] = 11). The grid labeled in this fashion represents RAY (I, J, 2). These labels are used in the main program to assign the proper equation to each node. The third time the grid is drawn, label each node with its material number, as determined above. This will represent RAY (I, J, 3).

A copy of the program, including SSDATA, is listed in Appendix A. Editing SSDATA to agree with the new problem will set up the data file when SSDATA is run.

When locating isotherms in the final temperature distribution, subroutine ISOTHM assumes a uniform internodal spacing. Thus if *semi-infinite* boundaries are used, edit ISOTHM as directed in the comment statements at the beginning of the subroutine to avoid error in interpolations.

When SSCONDUCT is run, it asks two questions of the user. The first is whether or not subroutine SSDATA should be run (it should be run if the formatted data file, STSDAT, is empty or if the user desires a change in that file). The second question is whether or not the user wants subroutine ISOTHM to be called to locate isotherms in the final temperature distribution. File STDOUT is the output file which lists the initial data and boundary conditions along with the final temperature distribution.

At the time of publication of this report, all boundary conditions are operational on all sides of the grid, except for the semi-infinite conditions, which are operational on the left edge, bottom, and right edge of the grid only.

**Instructions for FEHEAT**

The first step in using program FEHEAT is to divide the region to be modeled into triangular elements. Care must be taken here in order to obtain an accurate and economical solution. Points which will help are:

1. Use the smallest elements in the areas where you would expect the highest temperature gradients.

2. Use large elements where small gradients are suspected in order to minimize the number of elements and minimize computer core requirements and running time (i.e. cost).

3. Avoid the use of triangles with high aspect ratios, that is, triangles with sides varying greatly in length.

4. When several materials are present, approximate their boundaries as closely as possible with the element boundaries. Each element can only consist of one material in the model.

The next step is to assign numbers to all of the nodes. Great care must be taken here to ensure computational efficiency. The form of the matrix of equations which yields the solution is directly affected by the numbering of the nodes. The smaller the bandwidth of this matrix, the lower the computational effort required to solve it. To minimize the bandwidth, number the nodes such that, for the entire region, the maximum difference between any two node numbers of any element is as small as possible. For instance, it's better to have a maximum difference of 10 among the node numbers of several elements than to have one element be 15 greater than the smallest element and have a less than 10 difference among the remaining elements. The element or elements with the largest difference between node numbers determines the bandwidth which must be carried for the entire solution.

Once the nodes have been numbered the elements themselves may be numbered. They may be numbered in any sequence without affecting computation efficiency. Now we can construct the input data files. They are

ED = element data file
NPD = nodal point data file
BCT = constant temperature boundary conditions file
IHG = internal heat generation file
SHF = specified heat flux file
CBS = convective boundary segments file
QUAN = control data file
TIOT = desired isotherm temperatures.

Of these eight input data files only ED, NPD and QUAN are necessary unless the boundary conditions require the use of BCT, IHG, SHF or CBS and the desire for isotherms requires the use of TIOT.

The input data for ED, the element data file, is shown for a sample problem in Table 1. The node numbers for a particular element must be entered in counterclockwise order around the element. The material type must be assigned to each element and the material thermal conductivity defined in the program. Any consistent set of units may be used for the thermal conductivity, temperature, and position variables. The numbers in this file must be in the format (I5, 4I6).

NPD, the nodal point data file, is constructed exactly as shown in Table 2. The X and Y coordinates of each node must be determined from the drawing of the region. These are entered in the format (I5, 2F10.4).

Constant temperature boundary conditions are input in file BCT. The format of this file is (boundary condition number, node number, assigned temperature). These may be listed in any order of node numbers, although the boundary condition numbers must be consecutive. A typical BCT file might appear as:

| | | |
|---|---|---|
| 1 | 3 | 50.0000 |
| 2 | 7 | 100.0000 |
| 3 | 10 | 125.0000 |
| 4 | 2 | 150.0000 |
| 5 | 39 | 20.0000 |
| 6 | 8 | 75.0000 |
| . | . | . |
| . | . | . |
| . | . | . |

Thus, for instance, boundary condition 3 would be on node 10 and it would have an assigned temperature of 125. Again, the units of temperature need only be consistent with their use elsewhere. Headings on the printout indicate temperatures are in °F, but this may be ignored if °C are used for input throughout. The format for input in file BCT is (I5, I6, F10.4).

File IHG contains the boundary conditions for internal heat generation. The format of this file is (boundary condition number, element number, assigned heat generation rate). Similar to file BCT, these may be listed in any order of element numbers, but the boundary condition numbers must be consecutive. A typical file could appear as:

| | | |
|---|---|---|
| 1 | 5 | 20.0000 |
| 2 | 11 | 10.0000 |
| 3 | 6 | 17.5000 |
| 4 | 22 | 3.3333 |
| 5 | 16 | 20.0000 |
| 6 | 4 | 35.0000 |
| . | . | . |
| . | . | . |
| . | . | . |

For instance, boundary condition 5 would be on element 16 where the rate of internal heat generation would be 20. Consistent units must be used for heat generation rate. The format for the input in file IHG is identical to that for file BCT; it is (I5, I6, F10.4).

File SHF contains the boundary conditions of specified heat flux. An ideally insulated boundary has a heat flux of zero. If no heat flux is specified for a node on a physical boundary of the system, it becomes a zero heat flux boundary, provided, of course, that no other boundary condition is specified there. The format of this file is (boundary condition number, node number, specified heat flux). Again, the nodes may be listed in any order but the boundary condition numbers must be consecutive. Consistent units must be used. A typical SHF file is:

| | | |
|---|---|---|
| 1 | 6 | 5.0000 |
| 2 | 17 | 27.0000 |
| 3 | 24 | 6.6667 |
| 4 | 4 | 32.5000 |
| 5 | 5 | 50.0000 |

| | | |
|---|---|---|
| 6 | 87 | 9.5000 |
| . | . | . |
| . | . | . |
| . | . | . |

For example, boundary condition 2 would be on node 17 where the specified heat flux would be 27. Again, the format for input into file SHF is (I5, I6, F10.4).

File CBS contains boundary conditions for segments of the boundary where convection occurs. A boundary segment subject to convection is defined by the nodes at its end. The format of this file is (boundary condition number, node number at one end, node number at the other end, convective heat transfer coefficient, ambient temperature). The boundary segments may be listed in any order but the boundary condition number must be consecutive. Consistent units must be used for the convective heat coefficient and the ambient temperature. A typical CBS file is:

| | | | | |
|---|---|---|---|---|
| 1 | 6 | 7 | 1.0000 | 20.0000 |
| 2 | 27 | 3 | 1.2500 | 25.0000 |
| 3 | 12 | 20 | 2.3750 | 55.2750 |
| 4 | 21 | 20 | 1.0000 | 20.0000 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |

In this case, for example, convective boundary segment 3 would have node 12 at one end and node 20 at the other. The convective heat transfer coefficient there would be 2.375 and the ambient temperature 55.275. The format for file CBS is (3I5, 2F8.4).

The QUAN file contains control data for the problem. The format of the file is (NN, NE, MC, MJC, MC1, MC2, NIT), where

NN = total number of nodes
NE = total number of elements
MC = total number of fixed temperature boundary conditions
MJC = total number of elements with internal heat generation
MC1 = total number of nodes with specified heat flux
MC2 = total number of boundary segments subject to convection
NIT = number of isotherms desired.

A typical QUAN file is:

104 166 26 0 0 0 6.

Thus, this problem would have 104 nodes, 166 elements and 26 constant temperature boundary conditions, and 6 isotherms would be desired. The format of this file is (7I6).

The final file used is TIOT. This file contains the temperature of the isotherms which the user wishes to locate. The format is ($IT_1$, $IT_2$, $IT_3$, $IT_4$..., $IT_{NIT}$), where $IT_i$ is the temperature of the $i$th isotherm desired, $1 \leq i \leq$ NIT. A typical TIOT file is:

10.00
30.00
50.00
70.00
90.00
100.00

The format for file TIOT is (F8.2)

With the necessary files ready, program FEHEAT can be executed. The output is self explanatory, giving the resultant temperatures at each node. If isotherms are requested, a set of coordinate pairs which will form the isothermal line will also be output. Input data are printed out along with results as a means of checking to see that all data are properly read in.

If problems occur in the use of this program two likely areas should be checked first. These are:

1. All dimension and common statements must be made large enough for the problem. Currently, the dimensions are set for a maximum of 175 nodes, 300 elements and a bandwidth of 30.

2. Most often, mistakes are made in the input data, particularly in files ED and NPD. A very easy way to check these files is to write a short plotting program which would read first the coordinates of all the nodes from the NPD file and store them. Then the program would read in the nodes for each of the elements from the ED file and, using the coordinates of these nodes as obtained from NPD, plot each element individually on a composite plot. After all the elements are plotted it should be exactly as the original discretization of the region was. If any stray lines are found, element sides missing, or incorrect boundaries drawn, the problem in the input data should be easily located.

## CONCLUSIONS

Two computer programs have been developed to solve the steady-state heat conduction equation under a variety of boundary conditions. The accuracy is the same for each when modeling problems with rectangular boundaries and no semi-infinite boundary conditions are modeled.

SSCONDUCT, the finite difference program, is by far the ea :est to set up and run for a new problem.

FEHEAT, the finite element program, has the advantage of being able to use elements of varying size throughout the problem. This provides for a better definition of curved boundaries, and makes the problem cheaper to run (less computer time) if large elements are used where the temperature gradient is small.

## LITERATURE CITED

**Forsythe, G.E., M.A. Malcolm and C.G. Moler** (1977) *Computer methods for mathematical computations.* Englewood Cliffs, New Jersey: Prentice-Hall.

**Gerald, C.F.** (1978) *Applied numerical analysis,* 2nd ed. Reading, Massachusetts: Addison-Wesley Publishing Co.

**Huebner, K.H.** (1975) *The finite element method for engineers.* New York: John Wiley and Son, Inc.

**Martin, R.S. and J.H. Wilkinson** (1967) Solution of symmetric and unsymmetric band equations and the calculation of eigenvectors of band matrices. *Numerische Mathematik*, vol. 9, no. 4, p. 279-301.

**Pars, L.A.** (1962) *An introduction to the calculus of variations.* London: Heineman.

**Schechter, R.S.** (1967) *The variational method in engineering.* New York: McGraw-Hill Book Co.

**Segerlind, L.J.** (1976) *Applied finite element analysis.* New York: John Wiley and Sons, Inc.

**Zienkiewicz, O.C.** (1977) *The finite element method,* 3rd ed. New York: McGraw-Hill Book Co.

## APPENDIX A: FINITE DIFFERENCE PROGRAM—DOCUMENTATION AND SAMPLE INPUT AND OUTPUT

### Program SSCONDUCT

```
C        SSCONDUCT
C     THIS FORTRAN PROGRAM SOLVES FOR STEADY STATE 2-D TEMPERATURE DISTRIBUTION
C    RESULTING FROM CONDUCTION HEAT TRANSFER. BOUNDARY CONDITIONS MAY INCLUDE
C    CONSTANT TEMPERATURES, CONVECTIVE SURFACES, SEMI-INFINITE, AND CONSTANT
C    FLUX BOUNDARIES. THE GRID MAY CONTAIN MANY DIFFERENT CONDUCTIVITIES.

C    DATA FOR SSCONDUCT IS GATHERED BY SSDATA, WHICH PUTS IT INTO FILE STSDAT.
C    SEE SSDATA FOR AN EXPLANATION OF VARIABLES.
C     FOR DIMENSIONS: RMAT(Y*X,2X+1),RVCT(Y*X,1),RXL(Y*X,(X+1))
C
      IMPLICIT INTEGER(A,B,G,W,X,Y,Z)
      COMMON/M1I/ A,X,Y,NISO
      COMMON/M1R/ DS,DI,TMAX,TMIN,FI(4),THK(2),H(2),TISO(5),
     & RAY(100,100,3)
      COMMON/M3/ RMAT(10004,205),RVCT(10004,5),RXL(10004,105)
      DIMENSION IFLAG(4)
      CALL CONTRL(3,'STSDAT',5)
      CALL CONTRL(2,'STDOUT',6)
      WRITE(1,1)
 1    FORMAT(1X,'IF YOU EDITED THE DATA SUBROUTINE SINCE YOU LAST ',
     C ' RAN THIS, TYPE "1"',/,1X,'AND SSDATA WILL BE EXECUTED.',/,1X,
     C 'OTHERWISE, TYPE ".")
      READ(1,*,ERR=1111) IT
      GO TO 2
 1111 CONTINUE
      WRITE(1,3)
 3    FORMAT(1X,'ERROR ON INPUT FOR YOUR RESPONSE')
      GO TO 999
 2    CONTINUE
      IF(IT.EQ.2) GO TO 4
      CALL SSDATA
      GO TO 6
 4    CONTINUE
      READ(5,10) DS,DI
 10   FORMAT(1X,2F9.4)
      READ(5,20) A,X,Y,NISO
 20   FORMAT(1X,4I5)
      READ(5,30) (THK(L),L=1,A)
      READ(5,30) (H(L),L=1,A)
      READ(5,30) (TISO(B),B=1,NISO)
 30   FORMAT(1X,F9.4)
      READ(5,31) (FI(I),I=1,4)
 31   FORMAT(1X,4F9.4)
      READ(5,35) ((RAY(I,J,3),J=1,X),I=1,Y)
      READ(5,35) ((RAY(I,J,2),J=1,X),I=1,Y)
      READ(5,35) ((RAY(I,J,1),J=1,X),I=1,Y)
 35   FORMAT(1X,11F7.2)
 6    CONTINUE
C   WRITE DATA FOR RUN INTO STDOUT ***** **** ***
      WRITE(6,50)
      WRITE(1,50)
 50   FORMAT(1X,'    DATA FOR THIS RUN OF SSCONDUCT:',/,/ )
      WRITE(6,51) X
      WRITE(1,51) X
 51   FORMAT(1X,'X=',I3)
      WRITE(6,52) Y
      WRITE(1,52) Y
 52   FORMAT(1X,'Y=',I3)
      WRITE(6,54) A
      WRITE(1,54) A
 54   FORMAT(1X,'A=',I3)
      WRITE(6,56) NISO
      WRITE(1,56) NISO
 56   FORMAT(1X,'NISO=',I3)
      WRITE(6,8) DS
      WRITE(1,8) DS
 8    FORMAT(1X,'DS=',F7.3)
```

```
      WRITE(6,58) DI
      WRITE(1,58) DI
58    FORMAT(1X,'DI=',F9.4)
      WRITE(6,53)
      WRITE(1,53)
53    FORMAT(1X,'TISO(B),B=1,NISO:')
      WRITE(6,59) (TISO(B),B=1,NISO)
      WRITE(1,59) (TISO(B),B=1,NISO)
59    FORMAT(1X,F7.2)
      WRITE(6,55)
      WRITE(1,55)
55    FORMAT(1X,'THK(L),L=1,A:')
      WRITE(6,30) (THK(L),L=1,A)
      WRITE(1,30) (THK(L),L=1,A)
      WRITE(6,57)
      WRITE(1,57)
57    FORMAT(1X,'H(L),L=1,A:')
      WRITE(6,30) (H(L),L=1,A)
      WRITE(1,30) (H(L),L=1,A)
      DO 15 I=1,4
      WRITE(6,16) I,FI(I)
      WRITE(1,16) I,FI(I)
16    FORMAT(1X,'FI(',I1,')=',F9.4)
15    CONTINUE
      WRITE(6,65)
      WRITE(6,890) ((RAY(I,J,3),J=1,17),I=1,Y)
      WRITE(6,892)
      WRITE(6,893) ((RAY(I,J,3),J=18,X),I=1,Y)
      WRITE(6,66)
      WRITE(6,890) ((RAY(I,J,2),J=1,17),I=1,Y)
      WRITE(6,892)
      WRITE(6,893) ((RAY(I,J,2),J=18,X),I=1,Y)
      WRITE(6,67)
      WRITE(6,890) ((RAY(I,J,1),J=1,17),I=1,Y)
      WRITE(6,892)
      WRITE(6,893) ((RAY(I,J,1),J=18,X),I=1,Y)
65    FORMAT(/,1X,'RAY(I,J,3):')
66    FORMAT(/,1X,'RAY(I,J,2):')
67    FORMAT(/,1X,'RAY(I,J,1):')
      WRITE(6,68)
68    FORMAT(/,1X,'BOUNDARY CONDITIONS:')
C ***** **** ***
      DI2=2.*DI-1.
C    IBW=BANDWIDTH
      IBW=2*X+1
      KNT=X*Y
C     DO 60 K=1,KNT
C     DO 61 L=1,IBW
C     RMAT(K,L)=0
C     RVCT(K,1)=0
61    CONTINUE
60    CONTINUE
      DO 64 K=1,4
      IFLAG(K)=0.
64    CONTINUE
C
C
C LOOP I20,I21 TO FORM COEFFS OF EQUATIONS FOR EACH NODE
C   FORM MATRIX RMAT TO CONTAIN DIAGONAL ELTS ONLY
      KOUNT=0
      DO 201 I=1,Y
      DO 200 J=1,X
      KOUNT=KOUNT+1
      INDEX=RAY(I,J,3)
      IF(I.EQ.1) GO TO 11
      IND1=RAY((I-1),J,3)
      TK1=2./(1./THK(IND1)+1./THK(INDEX))
11    CONTINUE
      IF(J.EQ.1) GO TO 12
      IND2=RAY(I,(J-1),3)
      TK2=2./(1./THK(IND2)+1./THK(INDEX))
12    CONTINUE
      IF(I.EQ.Y) GO TO 13
      IND3=RAY((I+1),J,3)
      TK3=2./(1./THK(IND3)+1./THK(INDEX))
13    CONTINUE
      IF(J.EQ.X) GO TO 14
      IND4=RAY(I,(J+1),3)
      TK4=2./(1./THK(IND4)+1./THK(INDEX))
14    CONTINUE
      KKK=RAY(I,J,2)
      X1=X+1
```

```
      X2=X+2
      XN=(2*X)+1
      GO TO (101,102,103,104,105,106,107,108,109,110,111,
     & 112,113,114,115,116,117,118,119,120,121,122,123,124,
     & 125,126,127,128,129,130,131,132,133),KKK
C INTERIOR NODES
 101  CONTINUE
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1+TK2+TK3+TK4)
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=TK3
      RVCT(KOUNT,1)=0.
      GO TO 200
C   CONSTANT TEMP BOUNDARYS
 102  CONTINUE
      RMAT(KOUNT,X1)=1.
      RVCT(KOUNT,1)=RAY(I,J,1)
C       SET IFLAG FOR CONST BOUND
      IF (I.NE.1) GO TO 202
      IFLAG(1)=2
      GO TO 200
 202  CONTINUE
      IF(I.NE.Y) GO TO 203
      IFLAG(3)=2
      GO TO 200
 203  CONTINUE
      IF(J.NE.1) GO TO 204
      IFLAG(2)=2
      GO TO 200
 204  CONTINUE
      IFLAG(4)=2
      GO TO 200
C   CONVECTIVE SURFACE
C                 TOP
 104  CONTINUE
      IF(I.NE.1) GO TO 148
      IFLAG(1)=4
      CONS=2.*H(INDEX)*DS
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK2+2.*TK3+TK4+CONS)
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=2.*TK3
      RVCT(KOUNT,1)=-1*CONS*RAY(I,J,1)
      GO TO 200
 148  CONTINUE
C                 LEFT SIDE
      IF(J.NE.1) GO TO 149
      IFLAG(2)=4
      CONS=2.*DS*H(INDEX)
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X1)=-1.*(TK1+TK3+2.*TK4+CONS)
      RMAT(KOUNT,X2)=2.*TK4
      RMAT(KOUNT,XN)=TK3
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)
      GO TO 200
 149  CONTINUE
C             BOTTOM
      IF(I.NE.Y) GO TO 141
      IFLAG(3)=4
      CONS=2.*DS*H(INDEX)
      RMAT(KOUNT,1)=2.*TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,X1)=-1.*(2.*TK1+TK2+TK4+CONS)
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)
      GO TO 200
 141  CONTINUE
C              RIGHT SIDE
      IFLAG(4)=4
      CONS=2.*DS*H(INDEX)
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=2.*TK2
      RMAT(KOUNT,X1)=-1.*(TK1+TK3+2.*TK2+CONS)
      RMAT(KOUNT,XN)=TK3
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)
      GO TO 200
C   CONSTANT INTERIOR NODES
 105  CONTINUE
      WRITE(6,151) I,J,RAY(I,J,1)
 151  FORMAT(1X,'CONSTANT INTERIOR NODE RAY(',I2,',',I2,',','1)',
     & '=',F6.2)
      RMAT(KOUNT,X1)=1.
```

```
      RVCT(KOUNT,1)=RAY(I,J,1)
      GO TO 200
C   CONSTANT HEAT FLUX BOUNDARY
 103  CONTINUE
C        TOP
      IF(I.NE.1) GO TO 161
      IFLAG(1)=6
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*((TK2)+(2.*TK3)+(TK4))
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=2.*TK3
      RVCT(KOUNT,1)=-FI(1)*DS
      GO TO 200
 161  CONTINUE
C        BOTTOM
      IF(I.NE.Y) GO TO 162
      IFLAG(3)=6
      RMAT(KOUNT,1)=2.*TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*((2.*TK1)+(TK2)+(TK4))
      RMAT(KOUNT,X2)=TK4
      RVCT(KOUNT,1)=-FI(3)*DS
      GO TO 200
 162  CONTINUE
C        LEFT SIDE
      IF(J.NE.1) GO TO 163
      IFLAG(2)=6
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X1)=-1.*((TK1)+(TK3)+(2.*TK4))
      RMAT(KOUNT,X2)=2.*TK4
      RMAT(KOUNT,XN)=TK3
      RVCT(KOUNT,1)=-FI(2)*DS
      GO TO 200
 163  CONTINUE
C        RIGHT SIDE
      IFLAG(4)=6
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=2.*TK
      RMAT(KOUNT,X1)=-1.*((TK1)+(2.*TK2)+(TK3))
      RMAT(KOUNT,XN)=TK3
      RVCT(KOUNT,1)=-FI(4)*DS
      GO TO 200
 1 6  CONTINUE
C        SEMI-INFINITE BOUNDARY
      IF(I.NE.1) GO TO 1  0
C        TOP
      WRITE(1,1000) I,J
      GO TO 999
 1060 CONTINUE
      IF(J.NE.1) GO TO 1061
C        LEFT
      IFLAG(2)=6
      DIL=2.*DI-1.
      TKL=THK(INDEX)
      RMAT(KOUNT,1)=DI2*TK1
      RMAT(KOUNT,X1)=-1.*(TK4/DI+DI2*TK1+DI2*TK3
     1 +TKL/DI)
      RMAT(KOUNT,X2)=TK4/DI
      RMAT(KOUNT,XN)=DI2*TK3
      RVCT(KOUNT,1)=-1.*(TKL/DI)*RAY(I,J,1)
      GO TO 200
 1061 CONTINUE
      IF(I.NE.Y) GO TO 1062
C        BOTTOM
      IFLAG(3)=6
      DI2=2.*DI-1.
      TKB=THK(INDEX)
      RMAT(KOUNT,1)=TK1/DI
      RMAT(KOUNT,X)=DI2*TK2
      RMAT(KOUNT,X1)=-1.*(TK1/DI+DI2*TK2+DI2*TK4
     1 +TKB/DI)
      RMAT(KOUNT,X2)=DI2*TK4
      RVCT(KOUNT,1)=-1.*((TKB/DI)*RAY(I,J,1))
      GO TO 200
 1062 CONTINUE
C        RIGHT
C        RIGHT
      IFLAG(4)=6
      DI2=2.*DI-1.
      TKR=THK(INDEX)
      RMAT(KOUNT,1)=DI2*TK1
      RMAT(KOUNT,X)=TK2/DI
      RMAT(KOUNT,X1)=-1.*(TK2/DI+DI2*TK1+DI2*TK3+TKR/DI)
      RMAT(KOUNT,XN)=DI2*TK3
```

```
      RVCT(KOUNT,1)=-1.*(TKR/DI)*RAY(I,J,1)
      GO TO 200
 1000 FORMAT(1X,'RAY(',I2,',',I2,') HAS NO EQUATION')
 107  CONTINUE
C        NODE ADJACENT TO LEFT SIDE SEMI-INF BOUNDARY
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2/DI
      RMAT(KOUNT,X1)=-1.*(TK1+TK2/DI+TK3+TK4)
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=TK3
      RVCT(KUNT,1)=0.
      GO TO 200
 108  CONTINUE
C        NODE ADJACENT TO BOTTOM SEMI-INF BNDRY
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1+TK2+TK3/DI+TK4)
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=TK3/DI
      RVCT(KOUNT,1)=0.
      GO TO 200
 109  CONTINUE
C        NODE ADJACENT TO RIGHT SEMI-INFINITE BONDARY
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1+TK2+TK3+TK4/DI)
      RMAT(KOUNT,X2)=TK4/DI
      RMAT(KOUNT,XN)=TK3
      RVCT(KOUNT,1)=0.
      GO TO 200
 110  CONTINUE
C        NODE ADJACENT TO TOP SEMI-INFINITE BOUNDARY
      RMAT(KOUNT,1)=TK1/DI
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1/DI+TK2+TK3+TK4)
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=TK3
      RVCT(KOUNT,1)=0.
      GO TO 200
 111  CONTINUE
C          CONSTANT CORNER
      RMAT(KOUNT,X1)=1.
      RVCT(KOUNT,1)=RAY(I,J,1)
      WRITE(6,171) I,J,RAY(I,J,1)
 171  FORMAT(1X,'CONSTANT CORNER RAY(',I2,',',I2,',','1)=',
     *F8.2)
      GO TO 200
 112  CONTINUE
      IF(J.NE.1) GO TO 1120
      IF(I.NE.1) GO TO 1121
C          TOP LEFT CORNER CONST FLUX
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=TK3
      RMAT(KOUNT,X1) =-1.*(TK3+TK4)
      RVCT(KOUNT,1)=-.5*(FI(1)+FI(2))
      WRITE(6,207)
 207  FORMAT(1X,'TOP LEFT CORNER CONST FLUX')
      GO TO 200
 1121 CONTINUE
C          BOTTOM LEFT CORNER CONST FLUX
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X1)=-1.*(TK4+TK1)
      RVCT(KOUNT,1)=-.5*(FI(2)+FI(3))
      WRITE(6,208)
 208  FORMAT(1X,'BOTTOM LEFT CORNER CONST FLUX')
      GO TO 200
 1120 CONTINUE
      IF(I.NE.Y) GO TO 1122
C          BOTTOM RIGHT CORNER CONST FLUX
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1+TK2)
      RVCT(KOUNT,1)=-.5*(FI(3)+FI(4))
      WRITE(6,209)
 209  FORMAT(1X,'BOTTOM RIGHT CORNER CONST FLUX')
      GO TO 200
 1122 CONTINUE
C          TOP RIGHT CORNER CONST FLUX
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,XN)=TK3
      RMAT(KOUNT,X1)=-1.*(TK2+TK3)
      RVCT(KOUNT,1)=-.5*(FI(1)+FI(4))
      WRITE(6,210)
```

```
 210  FORMAT(1X,'TOP RIGHT CORNER CONST FLUX')
      GO TO 200
 113  CONTINUE
      IF(J.NE.1) GO TO 1130
      IF(I.NE.1) GO TO 1131
C           TOP LEFT CORNER CONVECTIVE
      CONS=2.*DS*H(INDEX)
      RMAT(KOUNT,X1)=-1.*(TK3+TK4+CONS)
      RMAT(KOUNT,XN)=TK3
      RMAT(KOUNT,X2)=TK4
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)
      WRITE(6,211)
 211  FORMAT(1X,'TOP LEFT CORNER CONVECTIVE')
      GO TO 200
 1131  CONTINUE
C           BOTTOM LEFT CORNER CONVECTIVE
      CONS=2.*DS*H(INDEX)
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X1)=-1.*(TK1+TK4+CONS)
      RMAT(KOUNT,X2)=TK4
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)
      WRITE(6,212)
 212  FORMAT(1X,'BOTTOM LEFT CORNER CONVECTIVE')
      GO TO 200
 1130 CONTINUE
      IF(I.NE.Y) GO TO 1132
C           BOTTOM RIGHT CORNER CONVECTIVE
      CONS=2.*DS*H(INDEX)
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1+TK2+CONS)
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)
      WRITE(6,213)
 213  FORMAT(1X,'BOTTOM RIGHT CORNER CONVECTIVE')
      GO TO 200
 1132 CONTINUE
C           TOP RIGHT CORNER CONVECTIVE
      CONS=2.*DS*H(INDEX)
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK2+TK3+CONS)
      RMAT(KOUNT,XN)=TK3
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)
      WRITE(6,214)
 214  FORMAT(1X,'TOP RIGHT CORNER CONVECTIVE')
      GO TO 200
 116  CONTINUE
C                CORNER: VERT=CONVECT,HORIZ=CONST FLUX
      IF(I.EQ.1) PHI=FI(1)
      IF(I.EQ.Y) PHI=FI(3)
      GO TO 1160
 117  CONTINUE
C              CORNER: HORIZ=CONVECT,VERT=CONST FLUX
      IF(J.EQ.1) PHI=FI(2)
      IF(J.EQ.X) PHI=FI(4)
 1160 CONTINUE
      IF(J.NE.1) GO TO 1161
      IF(I.NE.1) GO TO 1162
C         TOP LEFT CORNER CONVECT, CFLUX
      CONS=DS*H(INDEX)
      RMAT(KOUNT,XN)=TK3
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,X1)=-1.*(TK3+TK4+CONS)
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)-PHI*DS
      WRITE(6,215)
 215  FORMAT(1X,'TOP LEFT CORNER CONVECTIVE & C FLUX')
      GO TO 200
 1162 CONTINUE
C         BOTTOM LEFT CORNER CONVECT, C FLUX
      CONS=DS*H(INDEX)
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,X1)=-1.*(TK1+TK4+CONS)
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)-PHI*DS
      WRITE(6,216)
 216  FORMAT(1X,'BOTTOM LEFT CORNER CONVECTIVE & CONST FLUX')
      GO TO 200
 1161 CONTINUE
      IF(I.NE.Y) GO TO 1163
C               BOTTOM RIGHT CORNER CONVECT, C FLUX
      CONS=DS*H(INDEX)
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1+TK2+CONS)
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)
```

```
      WRITE(6,217)
  217 FORMAT(1X,'BOTTOM RIGHT CORNER CONVECT & C FLUX')
      GO TO 200
 1163 CONTINUE
C         TOP RIGHT CORNER ECNVECT,C FLUX
      CONS=DS*H(INDEX)
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,XN)=TK3
      RMAT(KOUNT,X1)=-1.*(TK2+TK3+CONS)
      RVCT(KOUNT,1)=-1.*CONS*RAY(I,J,1)-PHI*DS
      WRITE(6,218)
  218 FORMAT(1X,'TOP RIGHT CORNER CONVECT, CONST FLUX')
      GO TO 200
  114 CONTINUE
      IF(J.NE.1) GO TO 1140
      IF(I.NE.1) GO TO 1141
C                TOP LEFT CORNER SEMI-INFINITE ON BOTH SIDES
      WRITE(1,1000) I,J
      GO TO 998
 1141 CONTINUE
C                  BOTTOM LEFT CORNER SEMI-INFINITE ON BOTH SIDES
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X2)=TK4
      TKI=TK1
      TKL=TK1
      TKB=TK1
      TLFT=RAY(I,J,1)
      TBTM=RAY(I,J,1)
      DI=100.
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,X1)=-1.*(TK1+TK4+TKL+TKB)
      RVCT(KOUNT,1)=-TKL*TLFT-TKB*TBTM
      WRITE (6,221)
  221 FORMAT(1X,'BOTTOM LEFT CORNER SEMI-INFINITE')
      GO TO 200
 1140 CONTINUE
      IF(I.NE.Y) GO TO 1142
C              BOTTOM RIGHT CORNER SEMI-INFINITE ON BOTH SIDES
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2
      TKB=TK1
      TKR=TK1
      TRIT=RAY(I,J,1)
      TBTM=RAY(I,J,1)
      DI=100.
      RMAT(KOUNT,X1)=-1.*(TK1+TK2+TKR+TKB)
      RVCT(KOUNT,1)=-TKR*TRIT-TKB*TBTM
      WRITE(6,222)
  222 FORMAT(1X,'BOTTOM RIGHT CORNER SEMI-INFINITE')
      GO TO 200
 1142 CONTINUE
C            TOP RIGHT CORNER SEMI-INFINITE ON BOTH SIDES
      WRITE(1,1000) I,J
      GO TO 998
  115 CONTINUE
C      SQUARE INTERIOR NODE NEXT TO TWO SEMI-INF SIDES
      IF(J.NE.2) GO TO 1150
      IF(I.NE.2) GO TO 1151
C          TOP LEFT
      WRITE(1,1000) I,J
      GO TO 998
 1151 CONTINUE
C          BOTTOM LEFT
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2/DI
      RMAT(KOUNT,X1)=-1.*(TK1+TK2/DI+TK3/DI+TK4)
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=TK3/DI
      RVCT(KOUNT,1)=0.
C     WRITE(6,226) I,J
  226 FORMAT(1X,'RAY(',I2,',',I2,',2)=15')
      GO TO 200
 1150 CONTINUE
      YY=Y-1
      IF(I.NE.YY) GO TO 1152
C           BOTTOM RIGHT
      RMAT(KOUNT,1)=TK1
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1+TK2+TK3/DI+TK4/DI)
      RMAT(KOUNT,X2)=TK4/DI
      RMAT(KOUNT,XN)=TK3/DI
      RVCT(KOUNT,1)=0.
      GO TO 200
```

```
 1152 CONTINUE
C            TOP RIGHT
      WRITE(1,1000) I,J
      GO TO 998
  118 CONTINUE
C        BOTTOM RIGHT CORNER HORIZ SEMI-INNF,VERT CONST FLX
      DI2=2.*DI-1.
      TKB=TK1
      RMAT(KOUNT,1)=TK1/(DI*2.)
      RMAT(KOUNT,X)=TK2*DI2
      RMAT(KOUNT,X1)=-1.*(TK1/(DI*2.)+TK2*DI2+TKB/(DI*2.))
      RVCT(KOUNT,1)=-1.*(TKB/(DI*2.))*RAY(I,J,1)-FI(4)*DI2*DS
      GO TO 200
  119 CONTINUE
C     BOTTOM RIGHT SQUARE NEXT TO SEMI-INF,RT SIDE CONST FLX
      DI2=2.*DI-1.
      RMAT(KOUNT,1)=TK1/2.
      RMAT(KOUNT,X)=TK2
      RMAT(KOUNT,X1)=-1.*(TK1/2.+TK2+TK3/(DI*2.))
      RMAT(KOUNT,XN)=TK3/(DI*2.)
      RVCT(KOUNT,1)=-FI(4)*DS
      GO TO 200
  123 CONTINUE
C     BOTTOM LEFT SQUARE NET TO SEMI-INF,LEFT SIDE CONST FLX
      DI2=2.*DI-1.
      RMAT(KOUNT,1)=TK1/2.
      RMAT(KOUNT,X1)=-1.*(TK1/2.+TK4+TK3/(DI*2.))
      RMAT(KOUNT,X2)=TK4
      RMAT(KOUNT,XN)=TK3/(DI*2.)
      RVCT(KOUNT,1)=-FI(2)*DS
      GO TO 200
  122 CONTINUE
C    BOTTOM LEFT CORNER HORIZ SEMI-INF,VERT CONST FLX
      DI2=2.*DI-1.
      TKB=TK1
      RMAT(KOUNT,1)=TK1/(DI*2.)
      RMAT(KOUNT,X1)=-1.*(TK1/(DI*2.)+TK4*DI2+TKB/(DI*2.))
      RMAT(KOUNT,X2)=TK4*DI2
      RVCT(KOUNT,1)=-1.*(TKB/(DI*2.))*RAY(I,J,1)
      GO TO 200
  130 CONTINUE
C         SEMI-INF NODE ABOVE SEMI-INF CORNER NODE
      IF(J.NE.1) GO TO 1300
C                  LEFT SIDE
      TK3=DI/(((DI-.5)/THK(IND3))+(.5/THK(INDEX)))
      TK4=DI/(((DI-.5)/THK(INDEX))+(.5/THK(IND4)))
      RMAT(KOUNT,1)=TK1*DI2
      RMAT(KOUNT,X2)=TK4/DI
      RMAT(KOUNT,XN)=TK3*(DI2/DI)
      RMAT(KOUNT,X1)=-1.*(TK1*DI2+TK4/DI+TK3*DI2/DI+THK(INDEX)/DI)
      RVCT(KOUNT,1)=-1.*(THK(INDEX)/DI)*RAY(I,J,1)
      GO TO 200
 1300 CONTINUE
C                   RIGHT SIDE
      TK2=DI/(((DI-.5)/THK(INDEX))+(.5/THK(IND2)))
      TK3=DI/(((DI-.5)/THK(IND3))+(.5/THK(INDEX)))
      RMAT(KOUNT,1)=TK1*DI2
      RMAT(KOUNT,X)=TK2/DI
      RMAT(KOUNT,XN)=TK3*(DI2/DI)
      RMAT(KOUNT,X1)=-1.*(TK1*DI2+TK2/DI+TK3*DI2/DI+THK(INDEX)/DI)
      RVCT (KOUNT,1)=-1.*(THK(INDEX)/DI)*RAY(I,J,1)
      GO TO 200
  133 CONTINUE
C          SEMI-INF NODE TO RIGHT OF SEMI-INF CORNER NODE
      IF(I.NE.1) GO TO 1330
C                TOP
      WRITE(1,1000) I,J
      GO TO 998
 1330 CONTINUE
C                BOTTOM
      TK1=DI/(((DI-.5)/THK(INDEX))+(.5/THK(IND1)))
      TK2=DI/(((DI-.5)/THK(IND2))+(.5/THK(INDEX)))
      RMAT(KOUNT,1)=TK1/DI
      RMAT(KOUNT,X)=TK2*DI2/DI
      RMAT(KOUNT,X1)=-1.*(TK1/DI+TK2*DI2/DI+TK4*DI2+THK(INDEX)/DI)
      RMAT(KOUNT,X2)=TK4*DI2
      RVCT(KOUNT,1)=-1.*(THK(INDEX)/DI)*RAY(I,J,1)
      GO TO 200
  120 CONTINUE
  121 CONTINUE
  124 CONTINUE
  125 CONTINUE
  126 CONTINUE
  127 CONTINUE
```

```
 129   CONTINUE
 129   CONTINUE
 132   CONTINUE
 131   CONTINUE
       WRITE(1,1000) I,J
       GO TO 998
 200   CONTINUE
 201   CONTINUE
C ---- ----  -----  -----
C  LOOP TO WRITE BOUNDARY TYPES ONTO STDOUT
       DO 90 K=1,4
       IF (K.NE.1) GO TO 9001
       WRITE(6,9005)
 9005  FORMAT(1X,'TOP BOUNDARY')
       GO TO 9009
 9001  CONTINUE
       IF (K.NE.2) GO TO 9002
       WRITE(6,9006)
 9006  FORMAT(1X,'LEFT BOUNDARY')
       GO TO 9009
 9002  CONTINUE
       IF (K.NE.3) GO TO 9003
       WRITE(6,9007)
 9007  FORMAT(1X,'BOTTOM BOUNDARY')
       GO TO 9009
 9003  CONTINUE
       WRITE(6,9008)
 9008  FORMAT(1X,'RIGHT BOUNDARY')
 9009  CONTINUE
       IF(IFLAG(K).NE.1) GO TO 91
       WRITE(6,95)
       GO TO 90
 91    CONTINUE
       IF (IFLAG(K).NE.2) GO TO 93
       WRITE(6,97)
       GO TO 90
 93    CONTINUE
       IF (IFLAG(K).NE.3) GO TO 94
       WRITE(6,98)
       GO TO 90
 94    CONTINUE
       IF(IFLAG(K).NE.4) GO TO 92
       WRITE(6,96)
       GO TO 90
 92    CONTINUE
 95    FORMAT(1H+,'                       CONSTANT TEMPERATURE')
 97    FORMAT(1H+,'                       CONVECTIVE SURFACE')
 98    FORMAT(1H+,'                       CONSTANT HEAT FLUX')
 96    FORMAT(1H+,'                       SEMI-INFINITE')
 90    CONTINUE
C ----       ----    ----   ----   ----
       WRITE(6,72) IB
       WRITE(1,72) IB
 72    FORMAT(/,1X,'BANDWIDTH=',I5)
C SOLVE MATRIX KMAT
       NLC=X
       NUC=X
       N=Y*X
       IA=Y*X
       M=1
       IB=Y*X
       IJOB=0
       IER=678
       WRITE(1,80)
       WRITE(6,80)
 80    FORMAT(1X,' NLC, NUC,   N,  IA,    M,   IB,IJOB, IER:')
       WRITE(1,81) NLC,NUC,N,IA,M,IB,IJOB,IER
       WRITE(6,81) NLC,NUC,N,IA,M,IB,IJOB,IER
 81    FORMAT(1X,8I5)

       CALL BANDMX(N,NLC,NUC,IA,M,IB,IJOB,IER)
       WRITE(1,74) IER
       WRITE(6,74) IER
 74    FORMAT(/,1X,'IER=',I3,':',/,5X,'IF IER=129 , MATRIX IS SINGULAR',
     & /,5X,'IF IER=0, EVERYTHING IS OKAY.')
       IF(IER.EQ.129) GO TO 998
       K=0
       DO 310 I=1,Y
       DO 300 J=1,X
       K=K+1
       RAY(I,J,1)=RVCT(K,1)
 300   CONTINUE
 310   CONTINUE
       WRITE(6,891)
```

```
891   FORMAT(1X,'FINAL TEMPERATURE DISTRIBUTION:')
      WRITE(6,890) ((RAY(I,J,1),J=1,17),I=1,Y)
      WRITE(6,892)
      WRITE(6,893) ((RAY(I,J,1),J=18,X),I=1,Y)
890   FORMAT(1X,17F7.2)
892   FORMAT(/,1X,'..THE REST OF THE COLUMNS:')
893   FORMAT(1X,4F7.2)
      CALL ISOTHM
      GO TO 998
999   CONTINUE
      WRITE(1,9999)
9999  FORMAT(1X,'ERROR IN ASSIGNING NODAL LOCATIONS')
998   CONTINUE
      CALL CONTRL(4,'STSDAT',5)
      CALL CONTRL(4,'STDOUT',6)
      CALL EXIT
      END
C
C  ISOTHM FINDS ISOTHERMS FOR SSCONDUCT---------------------
      SUBROUTINE ISOTHM
      IMPLICIT INTEGER(A,B,Q,W,X,Y,Z)
      COMMON/M1I/ A,X,Y,NISO
      COMMON/M1R/ DS,DI,TMAX,TMIN,FI(4),THK(2),H(2),TISO(9),
     & RAY(100,100,3)
      DIMENSION EXRY(9,60), EYRY(9,60), COUNT(9)
      DIMENSION IX(5),IY(3)
      CALL CONTRL(2,'POINTS',7)
      CNTPT=0
      DO 5 L=1,9
      COUNT(L)=0
5     CONTINUE
C   CHANGE THE NXT TWO STMTS TO AGREE WITH DIMENSION:
      DO 6 K=1,40
      DO 7 B=1,9
      EXRY(B,K)=0
      EYRY(B,K)=0
7     CONTINUE
6     CONTINUE
C   TMAX=TEMP OF HOTTEST ISOTHERM
C   TMIN=TEMP OF COLDEST ISOTHERM
C   NISO= NO OF ISOTHERMS
C        LET 1= HOTTEST ISOTHERM
C   COUNT(B)=COUNTER FOR NO. ELTS IN EACH ISOTHERM
C   TISO(B)= TEMP OF EACH ISOTHERM
C   EXRY(B,K)=ARRAY FOR X-COORDINATES
C        B INDICATES WHICH ISOTHERM
C        K=COUNT(B) & INDICATES POSITION OF PT IN ISOTHERM LIST
C   LOOP TO LOCATE ISOTHERMS
C         FOR LEFT SEMI-INNF,LET XL=3
C         FOR BOTTOM SEMI-INF,LET YB=Y-3
C         FOR RIGHT SEMI-INF,LET XR=X-3, AND COMMENT OUT LOOP 800
C         FOR TOP SEMI-INF,LET YT=3, AND COMMENT OUT LOOP 860
C         IF NOT USING SEMI-INF,XL=1,XR=X-1,YT=2,YB=Y
      XL=3
      XR=X-1
      YB=Y-3
      YT=2
      DO 100 I=YT,YB
      DO 200 J=XL,XR
C
C EXAMINE TEMPS HORIZONTALLY
      RJ=RAY(I,J,1)
      RJ1=RAY(I,(J+1),1)
C     IF((RJ.GT.TISO(1)).AND.(RJ1.GT.TISO(1))) GO TO 500
C     IF((RJ.LT.TISO(NISO)).AND.(RJ1.LT.TISO(NISO))) GO TO 500
      DO 500 B=1,NISO
      IF((TISO(B).GT.RJ).AND.(TISO(B).LT.RJ1)) GO TO 400
      IF((TISO(B).LT.RJ).AND.(TISO(B).GT.RJ1)) GO TO 400
      IF(TISO(B).EQ.RJ) GO TO 402
      GO TO 500
400   CONTINUE
C     WRITE(7,401) I,J,RAY(I,J,1)
401   FORMAT(1X,'RAY(',I3,I3,',1)=',F6.2)
      EXCO=(RAY(I,J,1)-TISO(B))/(RAY(I,J,1)-RAY(I,(J+1),1))+J
      EXCO=DS*(EXCO-1)
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EXRY(B,K)=EXCO
      EYRY(B,K)=DS*(I-1)
      CNTPT=CNTPT+1
      GO TO 500
402   CONTINUE
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
```

```
      EXRY(B,K)=DS*(J-1)
      EYRY(B,K)=DS*(I-1)
      CNTPT=CNTPT+1
 500  CONTINUE
C EXAMINE TEMPS VERTICALLY
      RI=RAY(I,J,1)
      RI1=RAY((I-1),J,1)
      IF((RI.GT.TISO(1)).AND.(RI1.GT.TISO(1))) GO TO 525
      IF((RI.LT.TISO(NISO)).AND.(RI1.LT.TISO(NISO))) GO TO 525
      DO 525 B=1,NISO
      IF((TISO(B).GT.RI).AND.(TISO(B).LT.RI1)) GO TO 550
      IF((TISO(B).LT.RI).AND.(TISO(B).GT.RI1)) GO TO 550
      GO TO 525
 550  CONTINUE
C     WRITE(7,401) I,J,RAY(I,J,1)
      EYCO=(RAY((I-1),J,1)-TISO(B))/(RAY((I-1),J,1)-RAY(I,J,1))+(I-1)
      EYCO=DS*(EYCO-1)
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EYRY(B,K)=EYCO
      EXRY(B,K)=DS*(J-1)
      CNTPT=CNTPT+1
 525  CONTINUE
 200  CONTINUE
 100  CONTINUE
C  800 LOOP IS FOR RIGHT HAND SIDE
      DO 800 I=YT,YB
      DO 851 B=1,NISO
      RI=RAY(I,X,1)
      RI1=RAY((I-1),X,1)
      IF((RI.GT.(TISO(B))).AND.(RI1.LT.(TISO(B)))) GO TO 852
      IF((RI.LT.(TISO(B))).AND.(RI1.GT.(TISO(B)))) GO TO 852
      IF(RI.EQ.TISO(B)) GO TO 853
      GO TO 851
 852  CONTINUE
      EYCO=(I-1)+(RI1-TISO(B))/(RI1-RAY(I,X,1))
      EYCO=DS*(EYCO-1)
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EYRY(B,K)=EYCO
      EXRY(B,K)=DS*(X-1)
      CNTPT=CNTPT+1
      GO TO 851
 853  CONTINUE
      COUNT(B)=COUNT( )+1
      K=COUNT(B)
      EYRY(B,K)=DS*(I-1)
      EXRY(B,K)=DS*(X-1)
      CNTPT=CNTPT+1
 851  CONTINUE
 800  CONTINUE
C             860 LOOP IS FOR TOP ROW
      XX=X-1
      I=1
      DO 860 J=XL,XR
      RJ=RAY(1,J,1)
      RJ1=RAY(1,(J+1),1)
      DO 861 B=1,NISO
      IF((TISO(B).GT.RJ).AND.(TISO( ).LT. J1)) GO TO 862
      IF((TISO(B).LT.RJ).AND.(TISO( ).GT.RJ1)) GO TO 862
      IF(RJ.EQ.TISO(B)) GO TO 863
      GO TO 861
 862  CONTINUE
      EXCO=(RAY(I,J,1)-TISO(B))/(RAY(I,J,1)-RAY(I,(J+1),1))+J
      EXCO=DS*(EXCO-1)
      COUNT(B)=COUNT( )+1
      K=COUNT(B)
      EXRY(B,K)=EXCO
      EYRY(B,K)=0
      CNTPT=CNTPT+1
      GO TO 861
 863  CONTINUE
      COUNT(B)=COUNT( )+1
      K=COUNT(B)
      EXRY(B,K)=DS*(J-1)
      EYRY(B,K)=0
      CNTPT=CNTPT+1
 861  CONTINUE
 860  CONTINUE
      WRITE(7,92) CNTPT
      DO 666 B=1,NISO
      L=COUNT(B)
      WRITE(1,91) B,L
      WRITE(7,91) B,L
```

```
 91     FORMAT(1X,'ISOTHM #',I2,' HAS',I4,' POINTS')
        IF(L.EQ.0) GO TO 666
        WRITE(7,90) (EXRY(B,K),EYRY(B,K),K=1,L)
 90     FORMAT(1X,2F15.3)
 666    CONTINUE
        WRITE(1,93) CNTPT
 93     FORMAT(1X,'TOTAL NO. POINTS FOUND=',I5)
 92     FORMAT(1X,I5)
        CALL CONTRL(4,'POINTS',7)
        RETURN
        END
C ****************************************************
C SSDATA, DATA FOR SSCONDUCT
        SUBROUTINE SSDATA
        IMPLICIT INTEGER(A,B,Q,W,X,Y,Z)
        COMMON/M1I/ A,X,Y,NISO
        COMMON/M1R/ DS,DI,TMAX,TMIN,FI(4),THK(2),H(2),TISO(9),
       & RAY(100,100,3)
C      ALL UNITS ARE IN METERS,HOURS,CELSIUS, BUT
C      IF YOU USE ENGLISH UNITS,MAKE SURE THAT
C        THEY ARE CONSISTENT.
C     DS=DISTANCE BETWEEN NODES (M)
C     TSRF=SURFACE TEMP (OUTSIDE GRID) (INFINITE DIST AWAY)
C     TBTM=BOTTOM TEMP (UNDER GRID) (INFINITE DIST AWAY)
C     TRIT=TEMP TO RIGHT OF GRID (INFINITE DIST)
C     TLFT=TEMP TO LEFT OF GRID  (INFINITE DIST AWAY)
C     X= NO. OF GRID NODES HORIZONTALLY
C     Y= NO. OF GRID NODES VERTICALLLY
C     A= NO. OF DIFFERENT MATERIALS IN THE GRID
C     DI= DISTANCE HALFWAY TO INFINITY (DI IS THE MULTIPLE OF DS)
C     THK(L)=THERMAL CONDUCTIVITY (W/M*K)
C     H(L)=CONVECTION COEFFICIENT
C     FI(1)= HEAT FLUX FROM TOP OF GRID
C     FI(2)= HEAT FLUX FROM LEFT SIDE OF GRID
C     FI(3)= HEAT FLUX FROM BOTTOM OF GRID
C     FI(4)= HEAT FLUX FROM RIGHT SIDE
C     RAY(I,J,1)=PRESENT NODAL TEMP
C     RAY(I,J,2)=LOCATION TYPE:
C                  =1 INTERIOR NODE, FLUCTUATING TEMP
C                  =2 NODE ON CONSTANT TEMP BOUNDARY
C                  =3 ON CONSTANT FLUX BOUNDARY
C                  =4 ON CONVECTIVE SURFACE BOUNDARY
C                  =5 INTERIOR NODE, CONSTANT TEMP
C                  =6 SEMI-INFINITE BOUNDARY
C                  =7     NODE ADJACENT TO LEFT SEMI-INF BNDRY
C                  =8     NODE ADJACENT TO BOTTOM SEMI-INF BNDRY
C                  =9     NODE ADJACENT TO RIGHT SEMI-INF BNDRY
C                  =10    NODE ADJACENT TO TOP SEMI-INF BNDRY
C                  =11 CONSTANT CORNER NODE
C                  =12 CONST FLUX (ON BOTHE SIDES) CORNER NODE
C                  =13 CONVECTIVE(ON BOTH SIDES) CORNER NODE
C                  =14 SEMI-INF (ON BOTH SIDES) CORNER NODE
C                  =15     SQUARE NODE ADJACENT TO TWO SEMI-INF SIDES
C                  =30     SEMI-INF NODE ABOVE SEMI-INF CORNER NODE
C                  =31     SEMI-INF NODE TO LEFT OF SEMI-INF CORNER NODE
C                  =32     SEMI-INF NODE BELOW SEMI-INF CORNER NODE
C                  =33     SEMI-INF NODE TO RIGHT OF SEMI-INF CORNER NODE
C                  =16 CORNER WITH VERTICAL SIDE CONVECT,HORIZ SIDE CONST FLUX
C                  =17 CORNER WITH HORIZ SIDE CONVECT,VERT SIDE CONST FLUX
C                  =18 BOTTOM RIGHT CORNER. VERT=CONST FLUX,HORIZ=SEMI-INF
C                  =19     RIGHT SIDE CONST FLUX NODE ADJACENT TO BOTTOM SEMI-INF
C                  =20 TOP LEFT CORNER. VERT=SEMI-INF,HORIZ=CONST FLUX
C                  =21     TOP CONST FLUX NODE ADJACENT TO LEFT SEMI-INF
C                  =22 BOTTOM LEFT COENER. VERT=CONST FLUX,HORIZ=SEMI-INF
C                  =23     LEFT SIDE CONST FLUX ADJACENT TO BOTTOM SEMI-INF
C                  =24 BOTTOM RIGHT CORNER. VERT=CONVECT, HOEIZ=SEMI-INF
C                  =25     RIGHT SIDE CONVECT NODE ADJAC TO BOTTOM SEMI-INF
C                  =26 TOP LEFT CORNER. VERT=SEMI-INF,HORIZ=CONVECT
C                  =27     TOP CONVECT NODE ADJAC TO LEFT SEMI-INF
C                  =28 BOTTOM LEFT CORNER. VERT=CONVECT,HORIZ=SEMI-INF
C                  =29     LEFT SIDE CONVECT NODE ADJACENT TO BOTTOM SEMI-INF
C     RAY(I,J,3)= INDEX OF MATERIAL (RANGES FROM 1 TO A)
C     NISO= NO. OF ISOTHERMS TO BE PLOTTED
C     TMAX= TEMP OF HOTTEST ISOTHERM (*C)
C     TMIN= TEMP OF COLDEST ISOTHERM (*C)
C     TISO(L)= ARRAY CONTAINING ISOTHERM TEMPS
C             TISO(1) HAS THE HOTTEST ISOTHERM
C     COUNT(3)=COUNTER FOR NO.ELTS IN EACH ISOTHERM
C
C INITIALIZE VARIABLES
        DS=.1
        A=1
        X=21
        Y=31
```

```
      DI=50.
C     TSRF...TBTM ONLY MATTER FOR CONST OR CONVECTIVE OR SEMI-INF BOUNDARY.
C     FOR CONVECTIVE BNDRY,TSRF...TBTM INDICATES TEMP AWAY FROM GRID.
C     FOR CONSTANT BOUNDRY,TSRF...TBTM INDICATE THE TEMP OF THE BOUNDARY.
C     FOR SEMI-INF BOUNDRY,TSRF...TBTM INDICATE THE TEMP AT INFINITY.
      TSRF=10.0
      TLFT=10.0
      TRIT=10.0
      TBTM=10.0
C
C   THE FOLLOWING VALUES MATTER ONLY IF YOU INTEND TO LOCATE
C   AND PLOT THE ISOTHERMS OF THE RESULTING TEMPERATURE
C   DISTRIBUTION.   IF YOU DO NOT WISH TO RUN SUBROUTINE
C   ISOTHM, LEAVE THE FOLLOWING 3 LINES AS THEY ARE.
      TMAX=100.0
      TMIN=25
      NISO=4
C   INITIALIZE ISOTHERM TEMPS
C     IN ORDER, WITH TISO(1) HOTTEST
      TISO(1)=100.0
      TISO(2)=70.0
      TISO(3)=50.0
      TISO(4)=30.0
C
C      INITIALIZE FLUXES FROM BOUNDARIES:
C            SET TO ZERO IF NOT USED OR IF INSULATED.
      FI(1)=0.
      FI(2)=0.
      FI(3)=0.
      FI(4)=0.
C
C   DO NOT CHANGE THE FOLLOWING 11 LINES
      DO 5 J=1,X
      DO 6 I=1,Y
      DO 7 K=1,3
      RAY(I,J,K)=0
 7    CONTINUE
 6    CONTINUE
 5    CONTINUE
      DO 8 K=1,A
      THK(K)=0
      H(K)=0
 8    CONTINUE
C
C
C SET UP RAY(I,J,3) INDEX OF MATERIALS
C   THERE ARE A MATERIALS. LET THE SURROUNDING
C   MATERIAL BE THE *ATH* MATERIAL.  START
C   WITH *1*.
C   INDICATE RAY(I,J,3) FOR MATERIALS #1 TO #(A-1):
C
C THIS LOOP ASSIGNS THE *ATH* MATERIAL TO
C   THE REMAINING NODES. DO NOT CHANGE THE LOOP.
      DO 10 J=1,X
      DO 20 I=1,Y
      IF (RAY(I,J,3).NE.0) GO TO 20
      RAY(I,J,3)=A
 20   CONTINUE
 10   CONTINUE
C SET UP MATERIAL PROPERTIES FOR EACH MATERIAL.
      THK(A)=1.44
      H(A)=0
C
C
C      SET UP RAY(I,J,2) NODAL LOCATION TYPE
C        BOUNDARYS
C CHANGE ONLY THE "RAY(I,J,2)=_ " STATEMENT.
      YY=Y-1
C   TOP BOUNDARY
      DO 40 J=1,X
      RAY(1,J,2)=2
      RAY(1,J,1)=TSRF
 40   CONTINUE
C   BOTTOM BOUNDARY
      DO 50 J=1,X
      RAY(Y,J,2)=6
      RAY(Y,J,1)=TBTM
 50   CONTINUE
C   LEFT BOUNDARY
      DO 60 I=2,YY
      RAY(I,1,2)=6
      RAY(I,1,1)=TLFT
 60   CONTINUE
C   RIGHT BOUNDARY
```

```
      DO 70 I=2,YY
      RAY(I,X,2)=3
      RAY(I,X,1)=TRIT
 70   CONTINUE
C          CORNERS
C       ADJUST BOTH RAY(I,J,2) AND RAY(I,J,1)
C      TOP LEFT CORNER
      RAY(1,1,2)=11
      RAY(1,1,1)=10.00
C      BOTTOM LEFT CORNER
      RAY(Y,1,2)=14
      RAY(Y,1,1)=10.0
C      BOTTOM RIGHT CORNER
      RAY(Y,X,2)=18
      RAY(Y,X,1)=10.0
C      TOP RIGHT CORNER
      RAY(1,X,2)=11
      RAY(1,X,1)=10.0
C INTERIOR NODES GIVEN FLUCTUATING TEMP
C   DON'T CHANGE THIS LOOP.
      YY=Y-1
      XX=X-1
      DO 90 J=2,XX
      DO 80 I=2,YY
      RAY(I,J,2)=1
 80   CONTINUE
 90   CONTINUE
C INDICATE LOCATION OF ALL INTERIOR NODES OF CONST TEMP
C   SET RAY(I,J,2)=5 FOR LOCATION TYPE:
      RAY(10,X,2)=5
      RAY(11,X,2)=5
      RAY(11,(X-1),2)=5
      RAY(12,X,2)=5
C THE FOLLOWING IS FOR LEFT SIDE & BOTTOM SEMI-INFINITE.
C ADJUST THESE LOOPS AND NUMBERS TO PROVIDE FOR
C NODES ADJACENT TO SEMI-INFINITE BOUNDARY NODES.
      YY=Y-1
      DO 41 I=2,YY
      RAY(I,2,2)=7
 41   CONTINUE
      XX=X-1
      DO 42 J=3,XX
      RAY((Y-1),J,2)=9
 42   CONTINUE
C
      RAY((Y-1),1,2)=23
      RAY((Y-1),X,2)=19
      RAY((Y-1),2,2)=15
C
C SET UP RAY(I,J,1) ...NODAL TEMPERATURES
C IF CONST TEMP BOUNDARYS ARE NOT UNIFORM TEMP,
C (TSRF,TBTM,TLFT,TRIT,RESPECTIVELY),THEN INDICATE
C TEMPERATURES OF BOUNDARY NODES HERE. ALSO INDICATE
C TEMPERATURES OF CONSTANT INTERIOR NODES HERE.
      RAY(10,X,1)=100.0
      RAY(11,X,1)=100.0
      RAY(11,(X-1),1)=100.0
      RAY(12,X,1)=100.0
C              DO NOT CHANGE THE FOLLOWING 16 LINES.
      WRITE(5,131) DS,DI
 131  FORMAT(1X,2F9.4)
      WRITE(5,132) A,X,Y,NISO
 132  FORMAT(1X,4I3)
      WRITE(5,133)(THK(L),L=1,A)
      WRITE(5,133) (H(L),L=1,A)
      WRITE(5,133) (TISO(B),B=1,NISO)
 133  FORMAT(1X,F9.4)
      WRITE(5,134) (FI(I),I=1,4)
 134  FORMAT(1X,4F9.4)
      WRITE(5,151) ((RAY(I,J,3),J=1,X),I=1,Y)
      WRITE(5,151) ((RAY(I,J,2),J=1,X),I=1,Y)
      WRITE(5,151) ((RAY(I,J,1),J=1,X),I=1,Y)
 151  FORMAT(1X,11F7.2)
      RETURN
      END
C ****************************************
C
C
C
C    BANDMX
C-----------------------------------------------------------------------
C    USAGE                    - CALL BANDMX (A,N,NLC,NUC,IA,B,M,IB,IJOB,XL,
C                                 IER)
C    ARGUMENTS     A          - INPUT/OUTPUT MATRIX OF DIMENSION N BY
C                                 (NUC+NLC+1). SEE PARAMETER IJOB.
C                  N          - ORDER OF MATRIX A AND THE NUMBER OF ROWS IN
```

```
C                   N. (INPUT)
C            NLC    - NUMBER OF LOWER CODIAGONALS IN MATRIX A.
C                       (INPUT)
C            NUC    - NUMBER OF UPPER CODIAGONALS IN MATRIX A.
C                       (INPUT)
C            IA     - ROW DIMENSION OF MATRIX A EXACTLY AS
C                       SPECIFIED IN THE DIMENSION STATEMENT IN THE
C                       CALLING PROGRAM. (INPUT)
C            B      - INPUT/OUTPUT MATRIX OF DIMENSION N BY M.
C                       ON INPUT, B CONTAINS THE M RIGHT-HAND SIDES
C                       OF THE EQUATION AX = B. ON OUTPUT, THE
C                       SOLUTION MATRIX X REPLACES B. IF IJOB = 1,
C                       B IS NOT USED.
C            M      - NUMBER OF RIGHT HAND SIDES (COLUMNS IN B).
C                       (INPUT)
C            IB     - ROW DIMENSION OF MATRIX B EXACTLY AS
C                       SPECIFIED IN THE DIMENSION STATEMENT IN THE
C                       CALLING PROGRAM. (INPUT)
C            IJOB   - INPUT OPTION PARAMETER. IJOB = I IMPLIES WHEN
C                         I = 0, FACTOR THE MATRIX A AND SOLVE THE
C                           EQUATION AX = B. ON INPUT, A CONTAINS THE
C                           COEFFICIENT MATRIX OF THE EQUATION AX = B,
C                           WHERE A IS ASSUMED TO BE AN N BY N BAND
C                           MATRIX. A IS STORED IN BAND STORAGE MODE
C                           AND THEREFORE HAS DIMENSION N BY
C                           (NLC+NUC+1). ON OUTPUT, A IS REPLACED
C                           BY THE U MATRIX OF THE L-U DECOMPOSITION
C                           OF A ROWWISE PERMUTATION OF MATRIX A. U
C                           IS STORED IN BAND STORAGE MODE.
C                         I = 1, FACTOR THE MATRIX A. A CONTAINS THE
C                           SAME INPUT/OUTPUT INFORMATION AS IF
C                           IJOB = 0.
C                         I = 2, SOLVE THE EQUATION AX = B. THIS
C                           OPTION IMPLIES THAT BANDMX HAS ALREADY
C                           BEEN CALLED USING IJOB = 0 OR 1 SO THAT
C                           THE MATRIX A HAS ALREADY BEEN FACTORED.
C                           IN THIS CASE, OUTPUT MATRICES A AND XL
C                           MUST HAVE BEEN SAVED FOR REUSE IN THE
C                           CALL TO BANDMX.
C            XL     - WORK AREA OF DIMENSION N*(NLC+1). THE FIRST
C                       NLC*N LOCATIONS OF XL CONTAIN COMPONENTS OF
C                       THE L MATRIX OF THE L-U DECOMPOSITION OF A
C                       ROWWISE PERMUTATION OF A. THE LAST N
C                       LOCATIONS CONTAIN THE PIVOT INDICES.
C            IER    - ERROR PARAMETER. (OUTPUT)
C                     TERMINAL ERROR
C                       IER = 129 INDICATES THAT MATRIX A IS
C                         ALGORITHMICALLY SINGULAR. (SEE THE
C                         CHAPTER L PRELUDE).
C   REQUIRED SUBROUTINES-SUB1,SUB2
C-----------------------------------------------------------------------
      SUBROUTINE BANDMX (N,NLC,NUC,IA,M,IB,IJOB,IER)
      COMMON/M3/ A(10004,205),B(10004,5),XL(10004,105)
C                DIMENSION A(Y*X,NUC+NLC+1)    2-D
C                          XL(Y*X,(NLC+1))     1-D
C                          B(Y*X,1)            2-D
      DATA               ZERO/0./,ONE/1.0/
C                                  FIRST EXECUTABLE STATEMENT
      IER = 0
      JBEG = NLC+1
      NLC1 = JBEG
      IF (IJOB .EQ. 2) GO TO 80
      RN = N
C                                  RESTRUCTURE THE MATRIX
C                                  FIND RECIPROCAL OF THE LARGEST
C                                  ABSOLUTE VALUE IN ROW I
      I = 1
      NC = JBEG+NUC
      NN = NC
      JEND = NC
      IF (N .EQ. 1 .OR. NLC .EQ. 0) GO TO 25
    5 K = 1
      P = ZERO
      DO 10 J = JBEG,JEND
         A(I,K) = A(I,J)
         Q =  ABS(A(I,K))
         IF (Q .GT. P) P = Q
         K = K+1
   10 CONTINUE
      IF (P .EQ. ZERO) GO TO 135
      XL(I,NLC1) = ONE/P
      IF (K .GT. NC) GO TO 20
      DO 15 J = K,NC
         A(I,J) = ZERO
```

```
   15 CONTINUE
   20 I = I+1
      JBEG = JBEG-1
      IF (JEND-JBEG .EQ. N) JEND = JEND-1
      IF (I .LE. NLC) GO TO 5
      JBEG = I
      NN = JEND
   25 JEND = N-NUC
      DO 40 I = JBEG,N
         P = ZERO
         DO 30 J = 1,NN
            Q =  ABS(A(I,J))
            IF (Q .GT. P) P = Q
   30    CONTINUE
         IF (P .EQ. ZERO) GO TO 135
         XL(I,NLC1) = ONE/P
         IF (I .EQ. JEND) GO TO 37
         IF (I .LT. JEND) GO TO 40
         K = NN+1
         DO 35 J = K,NC
            A(I,J) = ZERO
   35    CONTINUE
   37    NN = NN-1
   40 CONTINUE
      L = NLC
C                                          L-U DECOMPOSITION
      DO 75 K = 1,N
         P =  ABS(A(K,1))*XL(K,NLC1)
         I = K
         IF (L .LT. N) L = L+1
         K1 = K+1
         IF (K1 .GT. L) GO TO 50
         DO 45 J = K1,L
            Q = ABS(A(J,1))*XL(J,NLC1)
            IF (Q .LE. P) GO TO 45
            P = Q
            I = J
   45    CONTINUE
   50    XL(I,NLC1) = XL(K,NLC1)
         XL(K,NLC1) = I
C                                          SINGULARITY FOUND
         Q = RN+P
         IF (Q .EQ. RN) GO TO 135
C                                          INTERCHANGE ROWS I AND K
         IF (K .EQ. I) GO TO 60
         DO 55 J = 1,NC
            P = A(K,J)
            A(K,J) = A(I,J)
            A(I,J) = P
   55    CONTINUE
   60    IF (K1 .GT. L) GO TO 75
         DO 70 I = K1,L
            P = A(I,1)/A(K,1)
            IK = I-K
            XL(K1,IK) = P
            DO 65 J = 2,NC
               A(I,J-1) = A(I,J)-P*A(K,J)
   65       CONTINUE
            A(I,NC) = ZERO
   70    CONTINUE
   75 CONTINUE
      IF (IJOB .EQ. 1) GO TO 9005
C                                          FORWARD SUBSTITUTION
   80 L = NLC
      DO 105 K = 1,N
         I = XL(K,NLC1)
         IF (I .EQ. K) GO TO 90
         DO 85 J = 1,M
            P = B(K,J)
            B(K,J) = B(I,J)
            B(I,J) = P
   85    CONTINUE
   90    IF (L .LT. N) L = L+1
         K1 = K+1
         IF (K1 .GT. L) GO TO 105
         DO 100 I = K1,L
            IK = I-K
            P = XL(K1,IK)
            DO 95 J = 1,M
               B(I,J) = B(I,J)-P*B(K,J)
   95       CONTINUE
  100    CONTINUE
  105 CONTINUE
C                                          BACKWARD SUBSTITUTION
```

```
      JBEG = NUC+NLC
      DO 125 J = 1,M
         L = 1
         K1 = N+1
         DO 120 I = 1,N
            K = K1-I
            P = B(K,J)
            IF (L .EQ. 1) GO TO 115
            DO 110 KK = 2,L
               IK = KK+K
               P = P-A(K,KK)*B(IK-1,J)
  110       CONTINUE
  115       B(K,J) = P/A(K,1)
            IF (L .LE. JBEG) L = L+1
  120    CONTINUE
  125 CONTINUE
      GO TO 9005
  135 IER = 129
 9000 CONTINUE
      WRITE(6,998)
  998 FORMAT(1X,'CALL SUB1')
      CALL SUB1(IER,'LEQT1B  ')
 9005 RETURN
      END
C
C
C
C   SUB1
C
C   ------------------------------------------------------------------------
C   PURPOSE             - PRINT A MESSAGE REFLECTING AN ERROR CONDITION
C   USAGE               - CALL SUB1 (IER,NAME)
C   ARGUMENTS     IER   - ERROR PARAMETER. (INPUT)
C                         IER = I+J WHERE
C                           I = 128 IMPLIES TERMINAL ERROR,
C                           I =  64 IMPLIES WARNING WITH FIX, AND
C                           I =  32 IMPLIES WARNING.
C                           J = ERROR CODE RELEVANT TO CALLING
C                               ROUTINE.
C                 NAME  - A SIX CHARACTER LITERAL STRING GIVING THE
C                         NAME OF THE CALLING ROUTINE. (INPUT)
C   REQUIRED SUBRTINE= SUB2
C   REMARKS        THE ERROR MESSAGE PRODUCED BY SUB1 IS WRITTEN
C                 ONTO THE STANDARD OUTPUT UNIT. THE OUTPUT UNIT
C                 NUMBER CAN BE DETERMINED BY CALLING SUB2 AS
C                 FOLLOWS..    CALL SUB2(1,NIN,NOUT).
C                 THE OUTPUT UNIT NUMBER CAN BE CHANGED BY CALLING
C                 SUB2 AS FOLLOWS..
C                                NIN = 0
C                                NOUT = NEW OUTPUT UNIT NUMBER
C                                CALL SUB2(3,NIN,NOUT)
C               SEE THE UGETIO DOCUMENT FOR MORE DETAILS.
C              -----------------------------------------------------------
      SUBROUTINE SUB1(IER,NAME)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER              IER
      REAL*8          NAME
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      REAL*8          NAMSET,NAMEQ
      DATA                 NAMSET/6HUERSET  /
      DATA                 NAMEQ/6H         /
C                                  FIRST EXECUTABLE STATEMENT
      DATA                 LEVEL/4/,IEQDF/0/,IEQ/1H=/
      IF (IER.GT.999) GO TO 25
      IF (IER.LT.-32) GO TO 55
      IF (IER.LE.128) GO TO 5
      IF (LEVEL.LT.1) GO TO 30
C                                  PRINT TERMINAL MESSAGE
      CALL SUB2(1,NIN,IOUNIT)
      IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAME
      IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAME
      GO TO 30
    5 IF (IER.LE.64) GO TO 10
      IF (LEVEL.LT.2) GO TO 30
C                                  PRINT WARNING WITH FIX MESSAGE
      CALL SUB2(1,NIN,IOUNIT)
      IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAME
      IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAME
      GO TO 30
   10 IF (IER.LE.32) GO TO 15
C                                  PRINT WARNING MESSAGE
      IF (LEVEL.LT.3) GO TO 30
      CALL SUB2(1,NIN,IOUNIT)
      IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAME
      IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAME
```

```
      GO TO 30
   15 CONTINUE
C                                   CHECK FOR UERSET CALL
      IF (NAME.NE.NAMSET) GO TO 25
      LEVOLD = LEVEL
      LEVEL = IER
      IER = LEVOLD
      IF (LEVEL.LT.0) LEVEL = 4
      IF (LEVEL.GT.4) LEVEL = 4
      GO TO 30
   25 CONTINUE
      IF (LEVEL.LT.4) GO TO 30
C                                   PRINT NON-DEFINED MESSAGE
      CALL SUB2(1,NIN,IOUNIT)
      IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAME
      IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAME
   30 IEQDF = 0
      GO TO 65
   35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3,
     &20H) FROM IMSL ROUTINE ,A6,A1,A6)
   40 FORMAT(36H *** WARNING WITH FIX ERROR  (IER = ,I3,
     &20H) FROM IMSL ROUTINE ,A6,A1,A6)
   45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,I3,
     &20H) FROM IMSL ROUTINE ,A6,A1,A6)
   50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I5,
     &20H) FROM IMSL ROUTINE ,A6,A1,A6)
C                                   SAVE P FOR P = R CASE
C                                     P IS THE PAGE NAME
C                                     R IS THE ROUTINE NAME
   55 IEQDF = 1
      NAMEQ = NAME
   65 CONTINUE
      RETURN
      END
C
C
C
C   SUB2
C-----------------------------------------------------------------------
C   PURPOSE               - TO RETRIEVE CURRENT VALUES AND TO SET NEW
C                             VALUES FOR INPUT AND OUTPUT UNIT
C                             IDENTIFIERS.
C
C   USAGE                 - CALL SUB2(IOPT,NIN,NOUT)
C
C   ARGUMENTS    IOPT     - OPTION PARAMETER. (INPUT)
C                             IF IOPT=1, THE CURRENT INPUT AND OUTPUT
C                             UNIT IDENTIFIER VALUES ARE RETURNED IN NIN
C                             AND NOUT, RESPECTIVELY.
C                             IF IOPT=2 (3) THE INTERVAL VALUE OF
C                             NIN (NOUT) IS RESET FOR SUBSEQUENT USE.
C                NIN      - INPUT UNIT IDENTIFIER.
C                             OUTPUT IF IOPT=1, INPUT IF IOPT=2.
C                NOUT     - OUTPUT UNIT IDENTIFIER.
C                             OUTPUT IF IOPT=1, INPUT IF IOPT=3.
C   REMARKS      EACH IMSL ROUTINE THAT PERFORMS INPUT AND/OR OUTPUT
C                OPERATIONS CALLS SUB2 TO OBTAIN TXE CURRENT UNIT
C                IDENTIFIER VALUES. IF SUB2 IS CALLED WITH IOPT=2 OR 3
C                NEW UNIT IDENTIFIER VALUES ARE ESTABLISHED. SUBSEQUENT
C                INPUT/OUTPUT IS PERFORMED ON THE NEW UNITS.
C-----------------------------------------------------------------------
C
      SUBROUTINE SUB2(IOPT,NIN,NOUT)
C                                  SPECIFICATIONS FOR ARGUMENTS
      IMPLICIT INTEGER*4(I-N)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      INTEGER            IOPT,NIN,NOUT
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            NIND,NOUTD
      DATA               NIND/5/,NOUTD/6/
C                                  FIRST EXECUTABLE STATEMENT
      IF (IOPT.EQ.3) GO TO 10
      IF (IOPT.EQ.2) GO TO 5
      IF (IOPT.NE.1) GO TO 9005
      NIN = NIND
      NOUT = NOUTD
      GO TO 9005
    5 NIND = NIN
      GO TO 9005
   10 NOUTD = NOUT
 9005 RETURN
      END
```

**Table A1. Sample output from SSCONDUCT — provides initial information and final temperature distribution.**

```
   DATA FOR THIS RUN OF SSCONDUCT:

X= 21
Y= 31
A=  1
NISO=   4
DS=   0.100
DI=   50.0000
TISO(B),B=1,NISO:
 100.00
  70.00
  50.00
  30.00
THK(L),L=1,A:
   1.4400
H(L),L=1,A:
   0.0000
FI(1)=    0.0000
FI(2)=    0.0000
FI(3)=    0.0000
FI(4)=    0.0000
RAY(I,J,3):
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
```

..THE REST OF THE COLUMNS:

| | | | |
|---|---|---|---|
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 |

RAY(I,J,2):

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 11.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 6.00 | 15.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 |
| 14.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |

..THE REST OF THE COLUMNS:

| | | | |
|---|---|---|---|
| 2.00 | 2.00 | 2.00 | 11.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 5.00 |
| 1.00 | 1.00 | 5.00 | 5.00 |
| 1.00 | 1.00 | 1.00 | 5.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 1.00 | 1.00 | 1.00 | 3.00 |
| 8.00 | 8.00 | 8.00 | 19.00 |
| 6.00 | 6.00 | 6.00 | 18.00 |

RAY(I,J,1):

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |

```
..THE REST OF THE COLUMNS:
  10.00   10.00   10.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00  100.00
   0.00    0.00  100.00  100.00
   0.00    0.00    0.00  100.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
   0.00    0.00    0.00   10.00
  10.00   10.00   10.00   10.00

BOUNDARY CONDITIONS:
CONSTANT CORNER RAY( 1, 1,1)= 10.00
CONSTANT CORNER RAY( 1,21,1)= 10.00
CONSTANT INTERIOR NODE RAY(10,21,1)=100.00
CONSTANT INTERIOR NODE RAY(11,20,1)=100.00
CONSTANT INTERIOR NODE RAY(11,21,1)=100.00
CONSTANT INTERIOR NODE RAY(12,21,1)=100.00


BOTTOM LEFT CORNER SEMI-INFINITE
TOP BOUNDARY     CONSTANT TEMPERATURE
LEFT BOUNDARY    SEMI-INFINITE
BOTTOM BOUNDARY SEMI-INFINITE
RIGHT BOUNDARY  CONSTANT HEAT FLUX

BANDWIDTH= 43
 NLC, NUC,   N,  IA,    M,  IB,IJOB, IER:
   21   21  651  651    1  651    0  478

IER=  0:
     IF IER=129, MATRIX IS SINGULAR
     IF IER=0, EVERYTHING IS OKAY.
```

FINAL TEMPERATURE DISTRIBUTION:

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| 10.04 | 11.92 | 11.97 | 12.05 | 12.14 | 12.26 | 12.43 | 12.57 | 12.77 | 12.99 | 13.24 | 13.53 | 13.85 | 14.19 | 14.55 | 14.92 | 15.28 |
| 10.08 | 13.81 | 13.93 | 14.08 | 14.27 | 14.50 | 14.78 | 15.12 | 15.50 | 15.95 | 16.46 | 17.03 | 17.66 | 18.36 | 19.09 | 19.85 | 20.60 |
| 10.12 | 15.67 | 15.84 | 16.06 | 16.35 | 16.69 | 17.11 | 17.60 | 18.18 | 18.84 | 19.60 | 20.46 | 21.42 | 22.48 | 23.62 | 24.80 | 25.98 |
| 10.16 | 17.48 | 17.70 | 17.99 | 18.36 | 18.82 | 19.36 | 20.01 | 20.77 | 21.64 | 22.65 | 23.80 | 25.09 | 26.53 | 28.09 | 29.75 | 31.44 |
| 10.19 | 19.21 | 19.48 | 19.84 | 20.29 | 20.85 | 21.52 | 22.31 | 23.24 | 24.32 | 25.56 | 26.99 | 28.61 | 30.44 | 32.47 | 34.68 | 36.99 |
| 10.23 | 20.86 | 21.17 | 21.59 | 22.12 | 22.77 | 23.55 | 24.47 | 25.56 | 26.82 | 28.29 | 29.98 | 31.93 | 34.16 | 36.69 | 39.50 | 42.56 |
| 10.26 | 22.40 | 22.76 | 23.23 | 23.83 | 24.56 | 25.43 | 26.47 | 27.69 | 29.12 | 30.78 | 32.72 | 34.97 | 37.57 | 40.59 | 44.07 | 48.02 |
| 10.29 | 23.84 | 24.23 | 24.75 | 25.40 | 26.20 | 27.16 | 28.29 | 29.63 | 31.19 | 33.01 | 35.15 | 37.64 | 40.58 | 44.05 | 48.17 | 53.08 |
| 10.32 | 25.15 | 25.58 | 26.13 | 26.83 | 27.69 | 28.71 | 29.91 | 31.33 | 32.99 | 34.93 | 37.21 | 39.88 | 43.06 | 46.86 | 51.48 | 57.22 |
| 10.34 | 26.33 | 26.79 | 27.38 | 28.11 | 29.09 | 30.07 | 31.32 | 32.79 | 34.51 | 36.52 | 38.87 | 41.63 | 44.91 | 48.85 | 53.68 | 59.78 |
| 10.37 | 27.39 | 27.86 | 28.47 | 29.23 | 30.15 | 31.24 | 32.52 | 34.01 | 35.74 | 37.76 | 40.11 | 42.86 | 46.09 | 49.94 | 54.60 | 60.33 |
| 10.38 | 28.31 | 28.80 | 29.42 | 30.20 | 31.12 | 32.22 | 33.50 | 34.98 | 36.70 | 38.68 | 40.96 | 43.60 | 46.67 | 50.24 | 54.43 | 59.36 |
| 10.40 | 29.09 | 29.59 | 30.23 | 31.01 | 31.93 | 33.02 | 34.27 | 35.72 | 37.38 | 39.28 | 41.45 | 43.92 | 46.73 | 49.92 | 53.54 | 57.57 |
| 10.41 | 29.75 | 30.26 | 30.90 | 31.67 | 32.58 | 33.64 | 34.86 | 36.25 | 37.84 | 39.62 | 41.64 | 43.89 | 46.41 | 49.19 | 52.22 | 55.44 |
| 10.42 | 30.28 | 30.80 | 31.43 | 32.19 | 33.05 | 34.10 | 35.27 | 36.60 | 38.08 | 39.74 | 41.58 | 43.61 | 45.82 | 48.20 | 50.71 | 53.27 |
| 10.42 | 30.70 | 31.22 | 31.84 | 32.58 | 33.44 | 34.42 | 35.53 | 36.77 | 38.16 | 39.68 | 41.34 | 43.15 | 45.07 | 47.09 | 49.16 | 51.19 |
| 10.42 | 31.01 | 31.53 | 32.14 | 32.85 | 33.68 | 34.61 | 35.66 | 36.81 | 38.09 | 39.47 | 40.97 | 42.56 | 44.22 | 45.93 | 47.63 | 49.26 |
| 10.42 | 31.23 | 31.73 | 32.33 | 33.02 | 33.81 | 34.69 | 35.67 | 36.74 | 37.91 | 39.16 | 40.50 | 41.89 | 43.33 | 44.78 | 46.19 | 47.51 |
| 10.41 | 31.35 | 31.85 | 32.43 | 33.09 | 33.84 | 34.57 | 35.58 | 36.57 | 37.64 | 38.77 | 39.96 | 41.19 | 42.43 | 43.66 | 44.84 | 45.92 |
| 10.39 | 31.39 | 31.88 | 32.45 | 33.08 | 33.79 | 34.57 | 35.42 | 36.34 | 37.31 | 38.33 | 39.39 | 40.46 | 41.54 | 42.60 | 43.59 | 44.48 |
| 10.38 | 31.36 | 31.84 | 32.39 | 33.00 | 33.67 | 34.41 | 35.20 | 36.04 | 36.92 | 37.85 | 38.79 | 39.74 | 40.68 | 41.59 | 42.44 | 43.19 |
| 10.35 | 31.27 | 31.74 | 32.27 | 32.86 | 33.49 | 34.18 | 34.92 | 35.69 | 36.51 | 37.34 | 38.19 | 39.03 | 39.86 | 40.65 | 41.37 | 42.01 |
| 10.33 | 31.12 | 31.59 | 32.10 | 32.66 | 33.26 | 33.91 | 34.60 | 35.32 | 36.06 | 36.82 | 37.59 | 38.35 | 39.09 | 39.77 | 40.40 | 40.95 |
| 10.30 | 30.93 | 31.38 | 31.88 | 32.42 | 32.99 | 33.61 | 34.25 | 34.92 | 35.61 | 36.30 | 37.00 | 37.68 | 38.34 | 38.95 | 39.51 | 39.99 |
| 10.27 | 30.69 | 31.13 | 31.62 | 32.13 | 32.69 | 33.27 | 33.87 | 34.50 | 35.14 | 35.78 | 36.42 | 37.04 | 37.64 | 38.19 | 38.69 | 39.11 |
| 10.23 | 30.41 | 30.85 | 31.32 | 31.82 | 32.35 | 32.90 | 33.48 | 34.07 | 34.67 | 35.27 | 35.86 | 36.43 | 36.98 | 37.48 | 37.93 | 38.32 |
| 10.19 | 30.11 | 30.54 | 30.99 | 31.48 | 31.99 | 32.52 | 33.07 | 33.63 | 34.20 | 34.76 | 35.32 | 35.85 | 36.36 | 36.83 | 37.24 | 37.60 |
| 10.14 | 29.77 | 30.19 | 30.64 | 31.11 | 31.60 | 32.12 | 32.65 | 33.18 | 33.73 | 34.27 | 34.80 | 35.30 | 35.78 | 36.22 | 36.62 | 36.95 |
| 10.09 | 29.41 | 29.82 | 30.26 | 30.72 | 31.20 | 31.70 | 32.21 | 32.74 | 33.26 | 33.78 | 34.29 | 34.78 | 35.24 | 35.67 | 36.04 | 35.37 |
| 10.04 | 10.06 | 10.08 | 10.10 | 10.12 | 10.14 | 10.16 | 10.17 | 10.19 | 10.20 | 10.21 | 10.23 | 10.24 | 10.25 | 10.25 | 10.26 | 10.27 |

..THE REST OF THE COLUMNS:

| | | | |
|---|---|---|---|
| 10.00 | 10.00 | 10.00 | 10.00 |
| 15.61 | 15.87 | 16.04 | 16.10 |
| 21.28 | 21.83 | 22.20 | 22.33 |
| 27.07 | 27.98 | 28.59 | 28.80 |
| 33.05 | 34.43 | 35.37 | 35.72 |
| 39.27 | 41.30 | 42.76 | 43.31 |
| 45.74 | 48.74 | 51.07 | 52.01 |
| 52.38 | 56.87 | 60.75 | 62.60 |
| 58.90 | 65.59 | 72.47 | 76.88 |
| 64.54 | 74.13 | 86.65 | 100.00 |
| 67.91 | 79.74 | 100.00 | 100.00 |
| 67.5[illegible] | 76.92 | 88.76 | 100.00 |
| 65.1[illegible] | 71.62 | 78.11 | 82.00 |
| 61.93 | 66.34 | 70.06 | 71.76 |
| 58.71 | 61.74 | 64.03 | 64.94 |
| 55.73 | 57.87 | 59.38 | 59.94 |
| 53.07 | 54.63 | 55.68 | 56.06 |
| 50.72 | 51.90 | 52.67 | 52.94 |
| 48.66 | 49.57 | 50.15 | 50.35 |
| 46.85 | 47.56 | 48.02 | 48.17 |
| 45.24 | 45.82 | 46.18 | 46.31 |
| 43.82 | 44.29 | 44.59 | 44.69 |
| 42.54 | 42.94 | 43.19 | 43.27 |
| 41.41 | 41.74 | 41.95 | 42.02 |
| 40.38 | 40.68 | 40.86 | 40.92 |
| 39.46 | 39.72 | 39.88 | 39.93 |
| 38.63 | 38.87 | 39.01 | 39.06 |
| 37.89 | 38.10 | 38.23 | 38.27 |
| 37.22 | 37.42 | 37.54 | 37.58 |
| 36.62 | 36.81 | 36.93 | 36.97 |
| 10.27 | 10.28 | 10.28 | 10.28 |

**Table A2. Sample output from SSCONDUCT—isotherm locations.**

```
   72
ISOTHM # 1 HAS    4 POINTS
              1.900              1.000
              2.000              0.900
              2.000              1.000
              2.000              1.100
ISOTHM # 2 HAS   12 POINTS
              1.864              0.800
              1.900              0.779
              1.757              0.900
              1.800              0.852
              1.718              1.000
              1.726              1.100
              1.775              1.200
              1.898              1.300
              1.800              1.231
              1.900              1.301
              2.000              0.752
              2.000              1.326
ISOTHM # 3 HAS   27 POINTS
              1.854              0.600
              1.900              0.587
              1.645              0.700
              1.700              0.664
              1.800              0.615
              1.537              0.600
              1.600              0.739
              1.468              0.900
              1.500              0.855
              1.424              1.000
              1.401              1.100
              1.393              1.200
              1.400              1.119
              1.402              1.300
              1.400              1.276
              1.427              1.400
              1.472              1.500
              1.542              1.600
              1.500              1.546
              1.651              1.700
              1.600              1.662
              1.700              1.735
              1.874              1.800
              1.800              1.781
              1.900              1.807
              2.000              0.577
              2.000              1.916
ISOTHM # 4 HAS   29 POINTS
              1.515              0.400
              1.600              0.374
              1.700              0.349
              1.800              0.331
              1.300              0.321
              1.276              0.500
              1.300              0.489
              1.400              0.444
              1.500              0.405
              1.161              0.600
              1.200              0.542
              0.953              0.700
              1.000              0.669
              1.100              0.601
              0.824              0.800
              0.900              0.743
              0.706              0.900
              0.800              0.822
              0.594              1.000
              0.600              0.995
              0.700              0.906
              0.484              1.100
              0.500              1.087
              0.375              1.200
              0.400              1.180
              0.264              1.300
              0.300              1.271
              0.200              1.361
              2.000              0.317
```

**Table A3. Sample input to SSCONDUCT—file generated by SSDATA (subroutine of SSCONDUCT).**

```
  0.1000   50.0000
1 21 31    4
  1.4400
  0.0000
100.0000
 70.0000
 50.0000
 30.0000
  0.0030    0.0000    0.0000    0.0000    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
  1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
```

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1.00 | 1.00 | | | | | | | | | |
| 11.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 |
| 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 2.00 | 11.00 | 6.00 |
| 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 5.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 5.00 | 5.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 5.00 | 6.00 |
| 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 |
| 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 |
| 7.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 7.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| 1.00 | 1.00 | 1.00 | 3.00 | 6.00 | 15.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 |
| 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 | 8.00 |
| 8.00 | 8.00 | 19.00 | 14.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |
| 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 | 6.00 |
| 6.00 | 18.00 | | | | | | | | | |
| 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 100.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100.00 | 100.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100.00 | 10.00 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 |
| 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 10.00 | 10.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| 0.00 | 0.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| 10.00 | 10.00 | | | | | | | | | |

## APPENDIX B: FINITE ELEMENT PROGRAM — DOCUMENTATION AND SAMPLE INPUT AND OUTPUT

### Program FEHEAT

```
C       THIS PROGRAM SOLVES 2 DIMENSIONAL STEADY STATE HEAT TRANSFER PROBLEMS
C       USING THE FINITE ELEMENT METHOD.  (ONLY TRIANGULAR ELEMENTS MAY BE USED)
C
C       DIMENSION THE MATRICES WHICH WILL NOT BE STORED IN COMMON STORAGE
      DIMENSION NMJ(300),NEC(300)
      DIMENSION BT(175)
C       DIMENSION THE REMAINING MATRICES INTO BLANK COMMON BLOCK STORAGE.
      COMMON/M1I/IMAT(300)
      COMMON/M1R/TKM(300),QI(300)
      COMMON/M12I/NC1(300),NC2(300)
      COMMON/M12R/X(175),Y(175),H(300),FQIE(175)
      COMMON/M2I/MC1,MC2,NN,NCASE,NBHF(175)
      COMMON/M2R/BHF(175),TA(300),RHST(175),RHS(300)
      COMMON/M23I/MC,NBT(175)
      COMMON/M23R/R1(175)
      COMMON/M3I/NE,NCN,NDF,NSZF,NBAND
      COMMON/M3R/TM(30,175)
      COMMON/M13I/NODE(300,3)
      COMMON/M13R/TME(3,3)
C       OPEN THE NECESSARY DATA FILES.
      CALL CONTRL(1,'NPDXXX',5)
      CALL CONTRL(1,'EDXXXX',6)
      CALL CONTRL(1,'BCTXXX',7)
      CALL CONTRL(1,'IHGXXX',8)
      CALL CONTRL(1,'SHFXXX',9)
      CALL CONTRL(1,'CHSXXX',14)
      CALL CONTRL(1,'QUANXX',11)
      CALL CONTRL(2,'OUTPUT',12)
      CALL CONTRL(1,'TIOTXX',13)
      READ(11,107,END=1005)NN,NE,MC,MJC,MC1,MC2,NIT
1005    CONTINUE
C       INITILIZE ALL MATRICES AND PARAMETERS TO ZERO.
      NCN=3
      NDF=1
      NCASE=1
      NSZF=NN*NDF
      TKM(1)=.833
      DO 2 I=1,NN,1
      X(I)=0.0
      Y(I)=0.0
      NBT(I)=0
      BT(I)=0.0
      NBHF(I)=0
      BHF(I)=0.0
      RHS(I)=0.0
      FQIE(I)=0.0
2         CONTINUE
      DO 28 J=1,NE,1
      NC1(J)=0
      NC2(J)=0
      H(J)=0.0
      TA(J)=0.0
      IMAT(J)=0
      NMJ(J)=0
      QI(J)=0.0
28          CONTINUE
C        READ IN THE NODAL POINT DATA.
      DO 3 J=1,NN,1
      READ(5,101,END=1000) I,X(I),Y(I)
3         CONTINUE
1000        CONTINUE
C        READ IN THE ELEMENT DATA.
      DO 4 I=1,NE,1
      READ(6,102,END=1001) J,NODE(J,1),NODE(J,2),NODE(J,3),IMAT(J)
4         CONTINUE
1001     CONTINUE
C        READ IN THE FIXED TEMPERATURE BOUNDARY CONDITIONS.
      IF(MC.EQ.0) GO TO 5
```

```
      DO 6 IM=1,MC,1
      READ(7,103,END=1002) M,NBT(M),BT(NBT(M))
6       CONTINUE
1002   CONTINUE
C       READ IN THE INTERNAL HEAT GENERATION VALUES FOR EACH ELEMENT.
5       IF(MJC.EQ.0) GO TO 1
      DO 7 IMJ=1,MJC,1
      READ(8,103) MJ,NMJ(MJ),QI(NMJ(MJ))
7       CONTINUE
C       READ IN THE HEAT FLUX VALUES FOR NODES WHERE HEAT FLUX IS SPECIFIED.
1       IF(MC1.EQ.0) GO TO 8
      DO 9 IMC1=1,MC1,1
      READ(9,103) M1,NBHF(M1),BHF(NBHF(M1))
9       CONTINUE
C       FOR BOUNDARY SEGMENTS SUBJECT TO CONVECTIVE HEAT TRANSFER READ
C       IN CORRESPONDING NODES ,H,AND AMBIENT TEMPERATURE.
8       IF(MC2.EQ.0) GO TO 10
      DO 11 M2=1,MC2,1
      READ(14,106) NEC(M2),NC1(NEC(M2)),NC2(NEC(M2)),H(NEC(M2)),
     X TA(NEC(M2))
11        CONTINUE
10        CONTINUE
      WRITE(12,120)NN,NE
      WRITE(12,121)
      DO 81 I=1,NN,1
      WRITE(12,101)I,X(I),Y(I)
81        CONTINUE
      WRITE(12,122)
      DO 82 J=1,NE,1
      WRITE(12,102)J,NODE(J,1),NODE(J,2),NODE(J,3),IMAT(J)
82        CONTINUE
      IF(MC)93,93,91
91        WRITE(12,123)
      DO 83 M=1,MC,1
      WRITE(12,103)M,NBT(M),BT(NBT(M))
83        CONTINUE
93        CONTINUE
      IF(MJC)94,94,92
92        WRITE(12,124)
      DO 84 I=1,MJC,1
      WRITE(12,103)MJ,NMJ(MJ),QI(NMJ(MJ))
84        CONTINUE
94        CONTINUE
      IF(MC1)95,95,97
97        WRITE(12,125)
      DO 85 M1=1,MC1,1
      WRITE(12,103)M1,NBHF(M1),BHF(NBHF(M1))
85        CONTINUE
95        CONTINUE
      IF(MC2)96,96,98
98        WRITE(12,126)
      DO 86 M2=1,MC2,1
      WRITE(12,126)NEC(M2),NC1(NEC(M2)),NC2(NEC(M2)),H(NEC(M2)),
     X TA(NEC(M2))
86        CONTINUE
96        CONTINUE
C       FIND BANDWIDTH FOR TRIANGULAR ELEMENTS.
      MX=0
      DO 66 I=1,NE,1
      L1=IABS(NODE(I,1)-NODE(I,3))
      L2=IABS(NODE(I,3)-NODE(I,2))
      L3=IABS(NODE(I,2)-NODE(I,1))
       MX=MAX0(MX,L1,L2,L3)
66    CONTINUE
      NBAND=NDF*(MX+1)
      WRITE(12,107)NBAND
      MUD=NBAND-1
      IO=0
      IOP=1
      CALL FORMK
      CALL FRHS(IT)
      CALL MCHB(RHS,TM,NSZF,1,MUD,IOP,0.5E-6,IER,IO)
      WRITE(12,127)
      DO 200 I=1,NN,1
      WRITE(12,130)I,RHS(I)
200         CONTINUE
      IF(NIT.EQ.0) GO TO 228
      CALL ISOTHM(RHS,NODE,X,Y,NE,NIT)
228         CONTINUE
101         FORMAT(I5,2F10.4)
102         FORMAT(I5,4I6)
103         FORMAT(I5,I6,F10.4)
106         FORMAT(3I6,2F10.4)
107         FORMAT(7I6)
```

```
109       FORMAT(5X,'THE BANDWIDTH IS',I6)
120       FORMAT(//,5X,'TOTAL NUMBER OF NODES =',I6,3X,
     X  'TOTAL NUMBER OF ELEMENTS = ',I6)
121       FORMAT(//,2X,'NODE',3X,'GLOBAL',4X,'GLOBAL',/,3X,'NO.',6X,'X',
     X 9X,'Y',1X,/,'--------------------------')
122       FORMAT(//,1X,'ELEMENT',1X,'NODE',2X,'NODE',2X,'NODE',2X,
     X 'MATL.',/,3X,'NO.',4X,'1',5X,'2',5X,'3',4X,'TYPE',/,1X,'-------',
     X '----------------------')
123       FORMAT(//,3X,'B.C.',2X,'NODE',2X,'TEMP',/,4X,'NO.',3X,'NO.',2X,
     X '(F)',/,1X,'-------------------------')
129       FORMAT(// 2X,'B.C.',2X,'NODE',6X,'QI',/,2X,'NO.',3X,'NO.',5X,
     X '(UNITS)')
125       FORMAT(//,2X,'B.C',2X,'NODE',6X,'HF',/,2X,'NO.',3X,'NO.',5X,
     X '(UNITS)')
126       FORMAT(//,1X,'SEGMENT',1X,'NODE',2X,'NODE',4X,'H',3X,'AMB',/,3X,
     X  'NO.',4X,'1',5X,'2',3X,'(UNITS)',2X,'TEMP(F)')
127       FORMAT(//,13X,'NODE',2X,'TEMPERATURE (F)',
     X  /,12X,'----------------------')
130       FORMAT(10X,I6,F13.2)
      CALL CONTRL(4,'NPDXXX',5)
      CALL CONTRL(4,'EDXXXX',6)
      CALL CONTRL(4,'BCTXXX',7)
      CALL CONTRL(4,'IHGXXX',8)
      CALL CONTRL(4,'SHFXXX',9)
      CALL CONTRL(4,'CBSXXX',14)
      CALL CONTRL(4,'QUANXX',11)
      CALL CONTRL(4,'OUTPUT',12)
      CALL CONTRL(4,'TIOTXX',13)
      CALL EXIT
      END
C
C     ------------------------------------------------------------------
      SUBROUTINE TSM(K)
      DIMENSION XC(3),YC(3),TK(300)
      COMMON/M1I/IMAT(300)
      COMMON/M1R/TKM(300),QI(300)
      COMMON/M12I/NC1(300),NC2(300)
      COMMON/M12R/X(175),Y(175),H(300),FQIE(175)
      COMMON/M13I/NODE(300,3)
      COMMON/M13R/TME(3,3)
C       K IS THE ELEMENT NUMBER.
      TK(K)=TKM(IMAT(K))
      N1=NODE(K,1)
      N2=NODE(K,2)
      N3=NODE(K,3)
C       DEFINE THE ELEMENT NODAL X AND Y GLOBAL COORDINATES.
      XC(1)=X(N1)
      YC(1)=Y(N1)
      XC(2)=X(N2)
      YC(2)=Y(N2)
      XC(3)=X(N3)
      YC(3)=Y(N3)
      A=1.0
      AA=1.0
C       DEFINE THE A'S,B'S,AND C'S OF THE LINEAR INTERPOLATION FUNCTIONS.
      B1=YC(2)-YC(3)
      B2=YC(3)-YC(1)
      B3=YC(1)-YC(2)
      C1=XC(3)-XC(2)
      C2=XC(1)-XC(3)
      C3=XC(2)-XC(1)
C       DETERMINE THE ELEMENT AREA
      DEL=ABS(0.5*(XC(1)*(YC(2)-YC(3))+XC(2)*(YC(3)-YC(1))+XC(3)
     X *(YC(1)-YC(2))))
C       FORM THE INFLUENCE MATRIX FOR TEMPERATURE.
      CONST=(TK(K)*A/(4.0*DEL))*AA
      TME(1,1)=(B1**2+C1**2)*CONST
      TME(1,2)=(B1*B2+C1*C2)*CONST
      TME(1,3)=(B1*B3+C1*C3)*CONST
      TME(2,1)=TME(1,2)
      TME(2,2)=(B2**2+C2**2)*CONST
      TME(2,3)=(B2*B3+C2*C3)*CONST
      TME(3,1)=TME(1,3)
      TME(3,2)=TME(2,3)
      TME(3,3)=(B3**2+C3**2)*CONST
      IF(NC1(K))315,330,340
340       KK1=NC1(K)
      KK2=NC2(K)
      IF(KK1.EQ.N1) K1=1
      IF(KK1.EQ.N2) K1=2
      IF(KK1.EQ.N3) K1=3
      IF(KK2.EQ.N1) K2=1
      IF(KK2.EQ.N2) K2=2
      IF(KK2.EQ.N3) K2=3
```

```
      IF(K1.EQ.K2) GO TO 315
      CONSTC=H(K)*(SQRT((X(NC1(K))-X(NC2(K)))**2+(Y(NC1(K))-
     X Y(NC2(K)))**2))
      TME(K1,K1)=TME(K1,K1)+CONSTC/3.
      TME(K1,K2)=TME(K1,K2)+CONSTC/6.
      TME(K2,K2)=TME(K2,K2)+CONSTC/3.
      TME(K2,K1)=TME(K2,K1)+CONSTC/6.
330       CONTINUE
C       FORM INFLUENCE MATRIX FOR INTERNAL HEAT GENERATION.
      QIE=QI(K)
      CONSTQ=DEL/12.
      QN=4.*QIE*CONSTQ
      FQIE(N1)=FQIE(N1)+QN
      FQIE(N2)=FQIE(N2)+QN
      FQIE(N3)=FQIE(N3)+QN
      GO TO 10
315       WRITE(12,320)
320       FORMAT(20X,'*******ERROR*******')
10       CONTINUE
      RETURN
      END
C     ---------------------- ------------- ----------------------------------
      SUBROUTINE FRHS(BT)
      DIMENSION BL(300),RBL(300)
      DIMENSION BT(175)
      COMMON/M12I/NC1(300),NC2(300)
      COMMON/M12R/X(175),Y(175),H(300),FQIE(175)
      COMMON/M2I/MC1,MC2,NN,NCASE,NBHF(175)
      COMMON/M2R/BHF(175),TA(300),RHST(175),RHS(300)
      COMMON/M23I/MC,NBT(175)
      COMMON/M23R/R1(175)
      DO 780 I=1,NN,1
      RHS(I)=FQIE(I)
780       CONTINUE
      IF(MC1)810,810,790
C       INSERT THE BOUNDARY HEAT FLUX.
790       DO 800 I=1,MC1,1
      RHS(NBHF(I))=RHS(NBHF(I))-BHF(NBHF(I))
800       CONTINUE
810       IF(MC2)840,840,820
C       ACCOUNT FOR CONVECTION AT THE BOUNDARY.
820       DO 830 I=1,MC2,1
      BL(I)=SQRT((X(NC1(I))-X(NC2(I)))**2+(Y(NC1(I))-Y(NC2(I)))**2)
      RBL(I)=2.*3.14159*((0.5*(X(NC1(I))+X(NC2(I)))))**2
      IF (NCASE.EQ.2) BL(I)=RBL(I)*BL(I)
      CTINF=H(I)*TA(I)*BL(I)/2.0
      RHS(NC1(I))=RHS(NC1(I))+CTINF
      RHST(NC1(I))=RHST(NC1(I))+CTINF
      RHS(NC2(I))=RHS(NC2(I))+CTINF
      RHST(NC2(I))=RHST(NC2(I))+CTINF
830       CONTINUE
840       CONTINUE
C       INSERT THE BOUNDARY CONDITIONS ON TEMPERATURE.
      DO 900 N=1,MC,1
      I=NBT(N)
      RHS(I)=R1(I)*BT(I)
900       CONTINUE
      RETURN
      END
C     -----------------------------------------------------------------------
      SUBROUTINE FORMK
C                         FORMS STIFFNESS MATRIX IN
C                         UPPER TRIANGULAR FORM
      COMMON/M23I/MC,NBT(175)
      COMMON/M23R/R1(175)
      COMMON/M3I/NE,NCN,NDF,NSZF,NBAND
      COMMON/M3R/TM(30,175)
      COMMON/M13I/NODE(300,3)
      COMMON/M13R/TME(3,3)
      DIMENSION ST(5250)
      EQUIVALENCE (ST(1),TM(1,1))
C
C                         ZERO STIFFNESS MATRIX
C
      DO 300 N=1,NSZF
      DO 300 M=1,NBAND
300     TM(M,N)=0
C                         SCAN ELEMENTS
C
      DO 400 N=1,NE
      CALL TSM(N)
C                         RETURN ESTIFM AS STIFFNESS MATRIX
C                         STORE ESTIFM IN SK
C                         FIRST ROWS
```

```
      DO 350 JJ=1,NCN
      NROWB =(NODE(N,JJ)-1)*NDF
      DO 350 J=1,NDF
      IF (NROWB) 350,305,305
305       NROWB =NROWB +1
      I=(JJ-1)*NDF+J
C                            THEN COLUMNS
C
      DO 330 KK=1,NCN
      NCOLB=(NODE(N,KK)-1)*NDF
      DO 320 K=1,NDF
      L=(KK-1)*NDF+K
      NCOL=NCOLB+K+1-NROWB
C                            SKIP STORING IF BELOW BAND
C
C
      IF(NCOL) 320,320,310
310       TM(NCOL,NROWB) = TM(NCOL,NROWB) + TME(I,L)
320       CONTINUE
330       CONTINUE
350       CONTINUE
400       CONTINUE
C                            INSERT BOUNDARY CONDITIONS
C
      DO 500 N=1,MC
      I=NBT(N)
      TM(1,I)=TM(1,I)*1.E15
      R1(I)=TM(1,I)
500       CONTINUE
      DO 1 J=1,NSZF
 169  FORMAT('ROW ',I2)
  166  FORMAT(10F10.2)
 167  FORMAT(//)
1       CONTINUE
C
C                     CHANGE STIFFNESS MATRIX TO CONSECUTIVE
C                                 STORAGE LOCATIONS
C
      M =1
      DO 777 I=1,NSZF
      K =NSZF-I +1
      DO 700 J =1,NBAND
      IF (J-K) 705,705,777
705       ST(M) =TM(J,I)
700       M =M +1
777       CONTINUE
 198   FORMAT(I6,6X,I6)
      RETURN
      END
C
C
C        -----------------------------------------------------------------------
      SUBROUTINE MCHB(R,A,M,N,MUD,IOP,EPS,IER,IO)
C          *   IBM SUBROUTINE MCHB(SEE IBM SSP BOOK). USED TO SOLVE MATRIX
C            EQUATION WHEN SQUARE MATRIX IS SYMMETRIC,BANDED,AND POSITIVE
C            DEFINITE. NO NEW INPUT DATA REQUIRED.
      DIMENSION R(300),A(5250)
      DOUBLE PRECISION TOL,SUM,PIV
      IF(IABS(IOP)-3) 1,1,43
1       IF(MUD) 45,2,2
2       MC=MUD+1
      IF(M-MC)46,3,3
3       MR=M-MUD
      IER=0
      IF(IOP)24,4,4
4       IEND=0
      LLDST=MUD
      DO 23 K=1,M
      IST=IEND+1
      IEND=IST+MUD
      J=K-MR
      IF(J)6,6,5
5       IEND=IEND-J
6       IF(J-1)8,8,7
7       LLDST=LLDST-1
8       LMAX=MUD
      J=MC-K
      IF(J)10,10,9
9       LMAX=LMAX-J
10        ID=0
      TOL=A(IST)*EPS
      DO 23 I=IST,IEND
      SUM=0.D0
      IF(LMAX)14,14,11
```

```
11        LL=IST
      LLD=LLDST
      DO 13 L=1,LMAX
      LL=LL-LLD
      LLL=LL+ID
      SUM=SUM+A(LL)*A(LLL)
      B=SUM
      B=ABS(B)
      IF(B.LT.1.0E-35) SUM = 0.0D0
      IF(LLD-MUD)12,13,13
12        LLD=LLD+1
13        CONTINUE
14        SUM=DBLE(A(I))-SUM
      IF(I-IST)15,15,20
15        IF(SUM)47,47,16
16        IF(SUM-TOL)17,17,19
17        IF(IER)18,18,19
18        IER=K-1
19        PIV=DSQRT(SUM)
      A(I)=PIV
      PIV=1.0D0/PIV
      GO TO 21
20        A(I)=SUM*PIV
21        ID=ID+1
      IF(ID-J)23,23,22
22        LMAX=LMAX-1
23        CONTINUE
      IF(IOP)24,44,24
24        ID=N*M
      IEND=IABS(IOP)-2
      IF(IEND)25,35,25
25        IST=1
      LMAX=0
      J=-MR
      LLDST=MUD
      DO 34 K=1,M
      PIV=A(IST)
      IF(PIV)26,48,26
26        PIV=1.0D0/PIV
      DO 30 I=K,ID,M
      SUM=0.0D0
      IF(LMAX)30,30,27
27        LL=IST
      LLL=I
      LLD=LLDST
      DO 29 L=1,LMAX
      LL=LL-LLD
      LLL=LLL-1
      SUM=SUM+A(LL)*R(LLL)
      IF(LLD-MUD)28,29,29
28        LLD=LLD+1
29        CONTINUE
30        R(I)=PIV*(DBLE(R(I))-SUM)
      IF(MC-K)32,32,31
31        LMAX=K
32        IST=IST+MC
      J=J+1
      IF(J)34,34,33
33        IST=IST-J
      LLDST=LLDST-1
34        CONTINUE
      IF(IEND)35,35,44
35        IST=M+(MUD*(M+M-MC))/2+1
      LMAX=0
      K=M
36        IEND=IST-1
      IST=IEND-LMAX
      PIV=A(IST)
      IF(PIV)37,49,37
37        PIV=1.0D0/PIV
      L=IST+1
      DO 40 I=K,ID,M
      SUM=0.0D0
      IF(LMAX)40,40,38
38        LLL=I
      DO 39 LL=L,IEND
      LLL=LLL+1
39        SUM=SUM+A(LL)*R(LLL)
40        R(I)=PIV*(DBLE(R(I))-SUM)
      IF(K-MR) 42,42,41
41        LMAX=LMAX+1
42        K=K-1
      IF(K)44,44,36
43        IER=-1
```

```
        GO TO 44
45         IER=-2
        GO TO 44
46         IER=-3
        GO TO 44
47         IER=-(K+10)
        WRITE (IO,9000) SUM,A(I)
9000          FORMAT(5X,D14.7,E14.7)
        GO TO 44
48         IER = -5
        GO TO 44
49         IER=-6
44         RETURN
        END
C       ----------------------------------------------------------------------------
        SUBROUTINE ISOTHM(T,NODE,XC,YC,NE,NIT)
        DIMENSION NODE(300,3),T(300),XC(175),YC(175)
        DIMENSION KK(10),TISO(10),X(100),Y(100),XS(100),YS(100)
        DO 350 I=1,NIT,1
        J=0
        READ(13,200,END=1008) TISO(I)
200        FORMAT(F8.2)
        TT=TISO(I)
        DO 57 K=1,NE,1
        N1=NODE(K,1)
        N2=NODE(K,2)
        N3=NODE(K,3)
        T1=T(N1)
        T2=T(N2)
        T3=T(N3)
        X1=XC(N1)
        Y1=YC(N1)
        X2=XC(N2)
        Y2=YC(N2)
        X3=XC(N3)
        Y3=YC(N3)
        TD1=T1-T2
        TD2=T1-T3
        TD3=T2-T3
        IF(TD1)8,60,9
 60     IF(ABS(T1-TT).GT..01) GO TO 28
        J=J+1
        X(J)=X1
        Y(J)=Y1
        J=J+1
        X(J)=X2
        Y(J)=Y2
        GO TO 28
8          TH1=T2
        TC1=T1
        IF(TT-TC1)28,10,10
10         IF(TH1-TT)28,11,11
11         J=J+1
        RATIO=(TH1-TT)/(TH1-TC1)
        X(J)=X2-((X2-X1)*RATIO)
        Y(J)=Y2-((Y2-Y1)*RATIO)
        GO TO 28
9          TH1=T1
        TC1=T2
        IF(TT-TC1)28,12,12
12         IF(TH1-TT)28,13,13
13         J=J+1
        RATIO=(TH1-TT)/(TH1-TC1)
        X(J)=X1-((X1-X2)*RATIO)
        Y(J)=Y1-((Y1-Y2)*RATIO)
28         IF(TD2)14,61,15
 61     IF(ABS(T1-TT).GT..01) GO TO 29
        J=J+1
        X(J)=X1
        Y(J)=Y1
        J=J+1
        X(J)=X3
        Y(J)=Y3
        GO TO 29
14         TH2=T3
        TC2=T1
        IF(TT-TC2)29,16,16
16         IF(TH2-TT)29,17,17
17         J=J+1
        RATIO=(TH2-TT)/(TH2-TC2)
        X(J)=X3-((X3-X1)*RATIO)
        Y(J)=Y3-((Y3-Y1)*RATIO)
        GO TO 29
15         TH2=T1
```

```
      TC2=T3
      IF(TT-TC2)29,18,18
18       IF(TH2-TT)29,19,19
19       J=J+1
      RATIO=(TH2-TT)/(TH2-TC2)
      X(J)=X1-((X1-X3)*RATIO)
      Y(J)=Y1-((Y1-Y3)*RATIO)
29       IF(TD3)20,62,21
 62   IF(ABS(T2-TT).GT..01) GO TO 30
      J=J+1
      X(J)=X2
      Y(J)=Y2
      J=J+1
      X(J)=X3
      Y(J)=Y3
      GO TO 30
20       TH3=T3
      TC3=T2
      IF(TT-TC3)30,22,22
22       IF(TH3-TT)30,23,23
23       J=J+1
      RATIO=(TH3-TT)/(TH3-TC3)
      X(J)=X3-((X3-X2)*RATIO)
      Y(J)=Y3-((Y3-Y2)*RATIO)
      GO TO 30
21       TH3=T2
      TC3=T3
      IF(TT-TC3)30,24,24
24       IF(TH3-TT)30,25,25
25       J=J+1
      RATIO=(TH3-TT)/(TH3-TC3)
      X(J)=X2-((X2-X3)*RATIO)
      Y(J)=Y2-((Y2-Y3)*RATIO)
30       CONTINUE
57       CONTINUE
      K=1
      XS(1)=X(1)
      YS(1)=Y(1)
      DO 80 M1=2,J,1
      KI=0
      DO 90 M2=1,K,1
      IF(ABS(XS(M2)-X(M1)).GT..01) GO TO 90
      IF(ABS(YS(M2)-Y(M1)).GT..01) GO TO 90
      KI=KI+1
90       CONTINUE
      IF(KI.NE.0) GO TO 80
      K=K+1
      XS(K)=X(M1)
      YS(K)=Y(M1)
80       CONTINUE
      KK(I)=K
      WRITE(12,210)TT
210        FORMAT(//,5X,'COORDINATES FOR THE',F8.2,' F ISOTHERM',/,4X,
     X  '-----------------------------------------')
      KK(I)=K
      DO 325 N=1,K,1
      WRITE(12,220)XS(N),YS(N)
325        CONTINUE
220        FORMAT(10X,2F10.4)
350        CONTINUE
1009    CONTINUE
      RETURN
      END
```

**Table B1. Sample output from FEHEAT — provides initial conditions and final temperature distribution.**

TOTAL NUMBER OF ELEMENTS = 166

TOTAL NUMBER OF NODES = 104

| NODE NO. | GLOBAL X | GLOBAL Y |
|---|---|---|
| 1 | 0.0000 | -0.1000 |
| 2 | 0.0383 | -0.0924 |
| 3 | 0.0707 | -0.0707 |
| 4 | 0.0924 | -0.0383 |
| 5 | 0.1000 | 0.0000 |
| 6 | 0.0924 | 0.0383 |
| 7 | 0.0707 | 0.0707 |
| 8 | 0.0383 | 0.0924 |
| 9 | 0.0000 | 0.1000 |
| 10 | 0.0000 | -0.1500 |
| 11 | 0.0574 | -0.1386 |
| 12 | 0.1061 | -0.1061 |
| 13 | 0.1386 | -0.0574 |
| 14 | 0.1500 | 0.0000 |
| 15 | 0.1386 | 0.0574 |
| 16 | 0.1061 | 0.1061 |
| 17 | 0.0574 | 0.1386 |
| 18 | 0.0000 | 0.1500 |
| 19 | 0.0000 | -0.2000 |
| 20 | 0.0765 | -0.1848 |
| 21 | 0.1414 | -0.1414 |
| 22 | 0.1848 | -0.0765 |
| 23 | 0.2000 | 0.0000 |
| 24 | 0.1848 | 0.0765 |
| 25 | 0.1414 | 0.1414 |
| 26 | 0.0765 | 0.1848 |
| 27 | 0.0000 | 0.2000 |
| 28 | 0.0000 | -0.3000 |
| 29 | 0.1148 | -0.2772 |
| 30 | 0.2121 | -0.2121 |
| 31 | 0.2772 | -0.1148 |
| 32 | 0.3000 | 0.0000 |
| 33 | 0.2772 | 0.1148 |
| 34 | 0.2121 | 0.2121 |
| 35 | 0.1148 | 0.2772 |
| 36 | 0.0000 | 0.3000 |
| 37 | 0.0000 | -0.4200 |
| 38 | 0.1607 | -0.3880 |
| 39 | 0.2970 | -0.2970 |
| 40 | 0.3880 | -0.1607 |
| 41 | 0.4200 | 0.0000 |
| 42 | 0.3880 | 0.1607 |
| 43 | 0.2970 | 0.2970 |
| 44 | 0.1607 | 0.3880 |
| 45 | 0.0000 | 0.4200 |
| 46 | 0.0000 | -0.5800 |
| 47 | 0.2220 | -0.5359 |
| 48 | 0.4101 | -0.4101 |
| 49 | 0.5359 | -0.2220 |
| 50 | 0.5800 | 0.0000 |
| 51 | 0.5359 | 0.2220 |
| 52 | 0.4101 | 0.4101 |
| 53 | 0.2220 | 0.5359 |
| 54 | 0.0000 | 0.5800 |
| 55 | 0.0000 | -0.7800 |
| 56 | 0.2985 | -0.7206 |
| 57 | 0.5515 | -0.5515 |
| 58 | 0.7206 | -0.2985 |
| 59 | 0.7800 | 0.0000 |
| 60 | 0.7206 | 0.2985 |
| 61 | 0.5515 | 0.5515 |
| 62 | 0.2985 | 0.7206 |
| 63 | 0.0000 | 0.7800 |
| 64 | 0.0000 | -1.0000 |
| 65 | 0.3827 | -0.9239 |
| 66 | 0.7071 | -0.7071 |
| 67 | 0.9239 | -0.3827 |
| 68 | 1.0000 | 0.0000 |
| 69 | 0.9239 | 0.3827 |
| 70 | 0.7071 | 0.7071 |
| 71 | 0.3827 | 0.9239 |
| 72 | 0.0000 | 1.0000 |
| 73 | 0.4142 | -1.0000 |
| 74 | 0.7071 | -1.0000 |
| 75 | 1.0000 | -1.0000 |
| 76 | 1.0000 | -0.7071 |
| 77 | 1.0000 | -0.4142 |
| 78 | 1.0000 | 0.4142 |
| 79 | 1.0000 | 0.7071 |
| 80 | 1.0000 | 1.0000 |
| 81 | 0.7071 | 1.0000 |
| 82 | 0.4142 | 1.0000 |
| 83 | 0.0000 | -1.4000 |
| 84 | 0.4000 | -1.4000 |
| 85 | 0.8000 | -1.4000 |
| 86 | 1.4000 | -1.4000 |
| 87 | 1.4000 | -1.0000 |
| 88 | 1.4000 | -0.6000 |
| 89 | 1.4000 | -0.2000 |
| 90 | 1.4000 | 0.2000 |
| 91 | 1.4000 | 0.6000 |
| 92 | 1.4000 | 1.0000 |
| 93 | 0.0000 | -2.0000 |
| 94 | 0.4000 | -2.0000 |
| 95 | 0.8000 | -2.0000 |
| 96 | 1.4000 | -2.0000 |
| 97 | 2.0000 | -2.0000 |
| 98 | 2.0000 | -1.4000 |
| 99 | 2.0000 | -1.0000 |
| 100 | 2.0000 | -0.6000 |
| 101 | 2.0000 | -0.2000 |
| 102 | 2.0000 | 0.2000 |
| 103 | 2.0000 | 0.6000 |
| 104 | 2.0000 | 1.0000 |

| ELEMENT NO. | NODE 1 | NODE 2 | NODE 3 | MATL. TYPE |
|---|---|---|---|---|
| 1 | 10 | 2 | 1 | 1 |
| 2 | 10 | 11 | 2 | 1 |
| 3 | 11 | 3 | 2 | 1 |
| 4 | 11 | 12 | 3 | 1 |
| 5 | 12 | 4 | 3 | 1 |
| 6 | 12 | 13 | 4 | 1 |
| 7 | 13 | 5 | 4 | 1 |
| 8 | 13 | 14 | 5 | 1 |
| 9 | 14 | 6 | 5 | 1 |
| 10 | 14 | 15 | 6 | 1 |
| 11 | 15 | 7 | 6 | 1 |
| 12 | 15 | 16 | 7 | 1 |
| 13 | 16 | 8 | 7 | 1 |
| 14 | 16 | 17 | 8 | 1 |
| 15 | 17 | 9 | 8 | 1 |
| 16 | 17 | 18 | 9 | 1 |
| 17 | 19 | 11 | 10 | 1 |
| 18 | 19 | 20 | 11 | 1 |
| 19 | 20 | 12 | 11 | 1 |
| 20 | 20 | 21 | 12 | 1 |
| 21 | 21 | 13 | 12 | 1 |
| 22 | 21 | 22 | 13 | 1 |
| 23 | 22 | 14 | 13 | 1 |
| 24 | 22 | 23 | 14 | 1 |
| 25 | 23 | 15 | 14 | 1 |
| 26 | 23 | 24 | 15 | 1 |
| 27 | 24 | 16 | 15 | 1 |
| 28 | 24 | 25 | 16 | 1 |
| 29 | 25 | 17 | 16 | 1 |
| 30 | 25 | 26 | 17 | 1 |
| 31 | 26 | 18 | 17 | 1 |
| 32 | 26 | 27 | 18 | 1 |
| 33 | 28 | 20 | 19 | 1 |
| 34 | 28 | 29 | 20 | 1 |
| 35 | 29 | 21 | 20 | 1 |
| 36 | 29 | 30 | 21 | 1 |
| 37 | 30 | 22 | 21 | 1 |
| 38 | 30 | 31 | 22 | 1 |

| | | | | |
|---|---|---|---|---|
| 39 | 31 | 23 | 22 | 1 |
| 40 | 31 | 32 | 23 | 1 |
| 41 | 32 | 24 | 23 | 1 |
| 42 | 32 | 33 | 24 | 1 |
| 43 | 33 | 25 | 24 | 1 |
| 44 | 33 | 34 | 25 | 1 |
| 45 | 34 | 26 | 25 | 1 |
| 46 | 34 | 35 | 26 | 1 |
| 47 | 35 | 27 | 26 | 1 |
| 48 | 35 | 36 | 27 | 1 |
| 49 | 37 | 29 | 28 | 1 |
| 50 | 37 | 38 | 29 | 1 |
| 51 | 38 | 30 | 29 | 1 |
| 52 | 38 | 39 | 30 | 1 |
| 53 | 39 | 31 | 30 | 1 |
| 54 | 39 | 40 | 31 | 1 |
| 55 | 40 | 32 | 31 | 1 |
| 56 | 40 | 41 | 32 | 1 |
| 57 | 41 | 33 | 32 | 1 |
| 58 | 41 | 42 | 33 | 1 |
| 59 | 42 | 34 | 33 | 1 |
| 60 | 42 | 43 | 34 | 1 |
| 61 | 43 | 35 | 34 | 1 |
| 62 | 43 | 44 | 35 | 1 |
| 63 | 44 | 36 | 35 | 1 |
| 64 | 44 | 45 | 36 | 1 |
| 65 | 46 | [illegible] | 37 | 1 |
| 66 | 46 | 47 | 38 | 1 |
| 67 | 47 | 39 | 38 | 1 |
| 68 | 47 | 48 | 39 | 1 |
| 69 | 48 | 40 | 39 | 1 |
| 70 | 48 | 49 | 40 | 1 |
| 71 | 49 | 41 | 40 | 1 |
| 72 | 49 | 50 | 41 | 1 |
| 73 | [illegible] | 42 | 41 | 1 |
| 74 | [illegible] | 51 | 42 | 1 |
| 75 | 51 | 43 | 42 | 1 |
| 76 | 51 | 52 | 43 | 1 |
| 77 | [illegible] | 44 | 43 | 1 |
| 78 | [illegible] | 53 | 44 | 1 |
| 79 | [illegible] | 45 | 44 | 1 |
| 80 | [illegible] | 54 | 45 | 1 |
| 81 | [illegible] | 47 | 46 | 1 |
| 82 | [illegible] | 56 | 47 | 1 |
| 83 | [illegible] | 48 | 47 | 1 |
| 84 | [illegible] | 57 | 48 | 1 |
| 85 | 57 | 49 | 48 | 1 |
| 86 | 57 | 58 | 49 | 1 |
| 87 | 58 | 50 | 49 | 1 |
| 88 | 58 | 59 | 50 | 1 |
| 89 | [illegible] | 51 | 50 | 1 |
| 90 | [illegible] | 60 | 51 | 1 |
| 91 | 60 | 52 | 51 | 1 |
| 92 | [illegible] | 61 | 52 | 1 |
| 93 | 61 | 53 | 52 | 1 |
| 94 | 61 | 62 | 53 | 1 |
| 95 | 62 | 54 | 53 | 1 |
| 96 | [illegible] | [illegible] | 54 | 1 |
| 97 | 64 | 56 | 55 | 1 |
| 98 | 64 | 65 | 56 | 1 |
| 99 | 65 | 57 | 56 | 1 |
| 100 | 65 | 66 | 57 | 1 |
| 101 | 66 | 58 | 57 | 1 |
| 102 | 66 | 67 | 58 | 1 |
| 103 | 67 | 59 | 58 | 1 |
| 104 | 67 | 68 | 59 | 1 |
| 105 | 68 | 60 | 59 | 1 |
| 106 | 68 | 69 | 60 | 1 |
| 107 | 69 | 61 | 60 | 1 |
| 108 | 69 | 70 | 61 | 1 |
| 109 | 70 | 62 | 61 | 1 |
| 110 | 70 | 71 | 62 | 1 |
| 111 | 71 | 63 | 62 | 1 |
| 112 | 71 | 72 | 63 | 1 |
| 113 | 64 | 73 | 65 | 1 |
| 114 | 73 | 66 | 65 | 1 |
| 115 | [illegible] | 77 | 67 | 1 |
| 116 | 77 | 69 | 67 | 1 |
| 117 | 68 | 78 | 69 | 1 |
| 118 | 76 | 79 | 69 | 1 |
| 119 | 70 | 82 | 71 | 1 |
| 120 | 82 | 72 | 71 | 1 |
| 121 | 73 | 74 | 66 | 1 |
| 122 | 74 | 75 | 66 | 1 |
| 123 | 75 | 76 | 66 | 1 |
| 124 | 76 | 77 | 66 | 1 |
| 125 | 78 | 79 | 70 | 1 |
| 126 | 79 | 80 | 70 | 1 |
| 127 | 80 | 81 | 70 | 1 |
| 128 | 81 | 82 | 70 | 1 |
| 129 | 83 | 84 | 64 | 1 |
| 130 | 84 | 73 | 64 | 1 |
| 131 | 84 | 74 | 73 | 1 |
| 132 | 84 | 85 | 74 | 1 |
| 133 | 85 | 75 | 74 | 1 |
| 134 | 85 | 86 | 75 | 1 |
| 135 | 86 | 87 | 75 | 1 |
| 136 | 75 | 87 | 76 | 1 |
| 137 | 87 | 88 | 76 | 1 |
| 138 | 76 | 88 | 77 | 1 |
| 139 | 88 | 89 | 77 | 1 |
| 140 | 77 | 89 | 68 | 1 |
| 141 | 89 | 90 | 68 | 1 |
| 142 | 68 | 90 | 76 | 1 |
| 143 | 90 | 91 | 78 | 1 |
| 144 | 78 | 91 | 79 | 1 |
| 145 | 91 | 92 | 79 | 1 |
| 146 | 79 | 92 | 80 | 1 |
| 147 | 93 | 94 | 83 | 1 |
| 148 | 94 | 84 | 83 | 1 |
| 149 | 94 | 95 | 84 | 1 |
| 150 | 95 | 85 | 84 | 1 |
| 151 | 95 | 96 | 85 | 1 |
| 152 | 96 | 86 | 85 | 1 |
| 153 | 96 | 97 | 86 | 1 |
| 154 | 97 | 98 | 86 | 1 |
| 155 | 86 | 98 | 87 | 1 |
| 156 | 98 | 99 | 87 | 1 |
| 157 | 87 | 99 | 88 | 1 |
| 158 | 99 | 100 | 88 | 1 |
| 159 | 88 | 100 | 89 | 1 |
| 160 | 100 | 101 | 89 | 1 |
| 161 | 89 | 101 | 102 | 1 |
| 162 | 89 | 102 | 90 | 1 |
| 163 | 90 | 102 | 103 | 1 |
| 164 | 90 | 103 | 91 | 1 |
| 165 | 91 | 103 | 104 | 1 |
| 166 | 91 | 104 | 92 | 1 |

| B.C. NO. | NODE NO. | TEMP (F) |
|---|---|---|
| 1 | 1 | 100.0000 |
| 2 | 2 | 100.0000 |
| 3 | 3 | 100.0000 |
| 4 | 4 | 100.0000 |
| 5 | 5 | 100.0000 |
| 6 | 6 | 100.0000 |
| 7 | 7 | 100.0000 |
| 8 | 8 | 100.0000 |
| 9 | 9 | 100.0000 |
| 10 | 80 | 10.0000 |
| 11 | 81 | 10.0000 |
| 12 | 82 | 10.0000 |
| 13 | 72 | 10.0000 |
| 14 | 93 | 10.0000 |
| 15 | 94 | 10.0000 |
| 16 | 95 | 10.0000 |
| 17 | 96 | 10.0000 |
| 18 | 97 | 10.0000 |
| 19 | 98 | 10.0000 |
| 20 | 99 | 10.0000 |
| 21 | 100 | 10.0000 |
| 22 | 101 | 10.0000 |
| 23 | 102 | 10.0000 |
| 24 | 103 | 10.0000 |
| 25 | 104 | 10.0000 |
| 26 | 92 | 10.0000 |

THE BANDWIDTH IS [illegible]3

| NODE | TEMPERATURE (F) |
|---|---|
| 1 | 100.00 |
| 2 | 100.00 |
| 3 | 100.00 |
| 4 | 100.00 |

| | |
|---|---|
| 5 | 100.00 |
| 6 | 100.00 |
| 7 | 100.00 |
| 8 | 100.00 |
| 9 | 100.00 |
| 10 | 87.52 |
| 11 | 87.48 |
| 12 | 87.36 |
| 13 | 87.17 |
| 14 | 86.92 |
| 15 | 86.61 |
| 16 | 86.31 |
| 17 | 85.10 |
| 18 | 86.02 |
| 19 | 78.66 |
| 20 | 78.59 |
| 21 | 78.40 |
| 22 | 76.05 |
| 23 | 77.59 |
| 24 | 77.05 |
| 25 | 76.51 |
| 26 | 76.08 |
| 27 | 75.92 |
| 28 | 66.38 |
| 29 | 66.27 |
| 30 | 65.97 |
| 31 | 65.41 |
| 32 | 64.63 |
| 33 | 63.65 |
| 34 | 62.64 |
| 35 | 61.80 |
| 36 | 61.47 |
| 37 | 56.31 |
| 38 | 56.19 |
| 39 | 55.79 |
| 40 | 55.06 |
| 41 | 53.97 |
| 42 | 52.54 |
| 43 | 50.91 |
| 44 | 49.50 |
| 45 | 48.91 |
| 46 | 46.81 |
| 47 | 46.69 |
| 48 | 46.26 |
| 49 | 45.37 |
| 50 | 43.97 |
| 51 | 42.00 |
| 52 | 39.53 |
| 53 | 37.10 |
| 54 | 36.02 |
| 55 | 38.21 |
| 56 | 34.16 |
| 57 | 37.77 |
| 58 | 35.78 |
| 59 | 35.12 |
| 60 | 32.63 |
| 61 | 29.02 |
| 62 | 24.92 |
| 63 | 22.86 |
| 64 | 31.01 |
| 65 | 31.15 |
| 66 | 30.89 |
| 67 | 29.86 |
| 68 | 26.03 |
| 69 | 25.25 |
| 70 | 20.40 |
| 71 | 13.79 |
| 72 | 10.00 |
| 73 | 28.93 |
| 74 | 25.52 |
| 75 | 21.63 |
| 76 | 25.00 |
| 77 | 27.71 |
| 78 | 23.05 |
| 79 | 15.86 |
| 80 | 10.00 |
| 81 | 10.00 |
| 82 | 10.00 |
| 83 | 21.24 |
| 84 | 20.33 |
| 85 | 18.13 |
| 86 | 14.15 |
| 87 | 16.72 |
| 88 | 18.72 |
| 89 | 19.37 |
| 90 | 18.02 |
| 91 | 14.66 |
| 92 | 10.00 |
| 93 | 10.00 |
| 94 | 10.00 |
| 95 | 10.00 |
| 96 | 10.00 |
| 97 | 10.00 |
| 98 | 10.00 |
| 99 | 10.00 |
| 100 | 10.00 |
| 101 | 10.00 |
| 102 | 10.00 |
| 103 | 10.00 |
| 104 | 10.00 |

COORDINATES FOR THE 10.00 F ISOTHERM

| | |
|---|---|
| 0.0000 | 1.0000 |
| 0.4142 | 1.0000 |
| 1.0000 | 1.0000 |
| 0.7071 | 1.0000 |
| 1.4000 | 1.0000 |
| 0.0000 | -2.0000 |
| 0.4000 | -2.0000 |
| 0.6000 | -2.0000 |
| 1.4000 | -2.0000 |
| 2.0000 | -2.0000 |
| 2.0000 | -1.7000 |
| 2.0000 | -1.0000 |
| 2.0000 | -0.6000 |
| 2.0000 | -0.2000 |
| 2.0000 | 0.2000 |
| 2.0000 | 0.6000 |
| 2.0000 | 1.0000 |

COORDINATES FOR THE 30.00 F ISOTHERM

| | |
|---|---|
| 0.5373 | 0.4830 |
| 0.5333 | 0.5363 |
| 0.5116 | 0.5476 |
| 0.2666 | 0.6436 |
| 0.1618 | 0.6562 |
| 0.0000 | 0.6714 |
| 0.8944 | -0.4268 |
| 0.9198 | -0.3810 |
| 0.9389 | 0.0000 |
| 0.8806 | 0.1276 |
| 0.7931 | 0.3285 |
| 0.2018 | -1.0000 |
| 0.3990 | -0.9633 |
| 0.5736 | -0.8406 |
| 0.7893 | -0.6249 |
| 0.7071 | -0.7558 |
| 0.7354 | -0.7354 |
| 0.7516 | -0.7071 |
| 0.0000 | -1.0415 |
| 0.0380 | -1.0380 |

COORDINATES FOR THE 50.00 F ISOTHERM

| | |
|---|---|
| 0.2088 | 0.3559 |
| 0.1588 | 0.3835 |
| 0.0000 | 0.4096 |
| 0.0547 | -0.5146 |
| 0.0000 | -0.5262 |
| 0.2006 | -0.4844 |
| 0.2493 | -0.4470 |
| 0.3657 | -0.3657 |
| 0.4007 | -0.3041 |
| 0.4652 | -0.1927 |
| 0.4735 | -0.1025 |
| 0.4835 | 0.0000 |
| 0.4450 | 0.1130 |
| 0.4237 | 0.1755 |
| 0.3215 | 0.2893 |
| 0.3061 | 0.3061 |

COORDINATES FOR THE 70.00 F ISOTHERM

| | |
|---|---|
| 0.0227 | -0.2658 |
| 0.0000 | -0.2705 |
| 0.1032 | -0.2493 |
| 0.1230 | -0.2355 |
| 0.1892 | -0.1892 |
| 0.2030 | -0.1668 |
| 0.2436 | -0.1009 |
| 0.2481 | -0.0715 |
| 0.2586 | 0.0000 |
| 0.2502 | 0.0331 |
| 0.2334 | 0.0967 |
| 0.2101 | 0.1279 |
| 0.1746 | 0.1746 |
| 0.1378 | 0.1971 |
| 0.0928 | 0.2241 |
| 0.0481 | 0.2324 |
| 0.0000 | 0.2410 |

COORDINATES FOR THE 90.00 F ISOTHERM

| | |
|---|---|
| 0.0076 | -0.1386 |
| 0.0536 | -0.1293 |
| 0.0987 | -0.0987 |
| 0.1284 | -0.0532 |
| 0.1382 | 0.0000 |
| 0.1269 | 0.0526 |
| 0.0966 | 0.0966 |
| 0.0520 | 0.1256 |
| 0.0413 | 0.1278 |
| 0.0000 | 0.1358 |

COORDINATES FOR THE 100.00 F ISOTHERM

| | |
|---|---|
| 0.0383 | -0.0924 |
| 0.0707 | -0.0707 |
| 0.0924 | -0.0383 |
| 0.0924 | 0.0383 |
| 0.0707 | 0.0707 |
| 0.0383 | 0.0924 |
| 0.0000 | 0.1000 |

**Table B2. Sample input to FEHEAT—all files must be typed in by user.**

**a. File ED.**

| | | | | |
|---|---|---|---|---|
| 1 | 10 | 2 | 1 | 1 |
| 2 | 10 | 11 | 2 | 1 |
| 3 | 11 | 3 | 2 | 1 |
| 4 | 11 | 12 | 3 | 1 |
| 5 | 12 | 4 | 3 | 1 |
| 6 | 12 | 13 | 4 | 1 |
| 7 | 13 | 5 | 4 | 1 |
| 8 | 13 | 14 | 5 | 1 |
| 9 | 14 | 6 | 5 | 1 |
| 10 | 14 | 15 | 6 | 1 |
| 11 | 15 | 7 | 6 | 1 |
| 12 | 15 | 16 | 7 | 1 |
| 13 | 16 | 8 | 7 | 1 |
| 14 | 16 | 17 | 8 | 1 |
| 15 | 17 | 9 | 8 | 1 |
| 16 | 17 | 18 | 9 | 1 |
| 17 | 19 | 11 | 10 | 1 |
| 18 | 19 | 20 | 11 | 1 |
| 19 | 20 | 12 | 11 | 1 |
| 20 | 20 | 21 | 12 | 1 |
| 21 | 21 | 13 | 12 | 1 |
| 22 | 21 | 22 | 13 | 1 |
| 23 | 22 | 14 | 13 | 1 |
| 24 | 22 | 23 | 14 | 1 |
| 25 | 23 | 15 | 14 | 1 |
| 26 | 23 | 24 | 15 | 1 |
| 27 | 24 | 16 | 15 | 1 |
| 28 | 24 | 25 | 16 | 1 |
| 29 | 25 | 17 | 16 | 1 |
| 30 | 25 | 26 | 17 | 1 |
| 31 | 26 | 18 | 17 | 1 |
| 32 | 26 | 27 | 18 | 1 |
| 33 | 28 | 20 | 19 | 1 |
| 34 | 28 | 29 | 20 | 1 |
| 35 | 29 | 21 | 20 | 1 |
| 36 | 29 | 30 | 21 | 1 |
| 37 | 30 | 22 | 21 | 1 |
| 38 | 30 | 31 | 22 | 1 |
| 39 | 31 | 23 | 22 | 1 |
| 40 | 31 | 32 | 23 | 1 |
| 41 | 32 | 24 | 23 | 1 |
| 42 | 32 | 33 | 24 | 1 |
| 43 | 33 | 25 | 24 | 1 |
| 44 | 33 | 34 | 25 | 1 |
| 45 | 34 | 26 | 25 | 1 |
| 46 | 34 | 35 | 26 | 1 |
| 47 | 35 | 27 | 26 | 1 |
| 48 | 35 | 36 | 27 | 1 |
| 49 | 37 | 29 | 28 | 1 |
| 50 | 37 | 38 | 29 | 1 |
| 51 | 38 | 30 | 29 | 1 |
| 52 | 38 | 39 | 30 | 1 |
| 53 | 39 | 31 | 30 | 1 |
| 54 | 39 | 40 | 31 | 1 |
| 55 | 40 | 32 | 31 | 1 |
| 56 | 40 | 41 | 32 | 1 |
| 57 | 41 | 33 | 32 | 1 |
| 58 | 41 | 42 | 33 | 1 |
| 59 | 42 | 34 | 33 | 1 |
| 60 | 42 | 43 | 34 | 1 |
| 61 | 43 | 35 | 34 | 1 |
| 62 | 43 | 44 | 35 | 1 |
| 63 | 44 | 36 | 35 | 1 |
| 64 | 44 | 45 | 36 | 1 |
| 65 | 46 | 38 | 37 | 1 |
| 66 | 46 | 47 | 38 | 1 |
| 67 | 47 | 39 | 38 | 1 |
| 68 | 47 | 48 | 39 | 1 |
| 69 | 48 | 40 | 39 | 1 |
| 70 | 48 | 49 | 40 | 1 |
| 71 | 49 | 41 | 40 | 1 |
| 72 | 49 | 50 | 41 | 1 |
| 73 | 50 | 42 | 41 | 1 |
| 74 | 50 | 51 | 42 | 1 |
| 75 | 51 | 43 | 42 | 1 |
| 76 | 51 | 52 | 43 | 1 |
| 77 | 52 | 44 | 43 | 1 |
| 78 | 52 | 53 | 44 | 1 |
| 79 | 53 | 45 | 44 | 1 |
| 80 | 53 | 54 | 45 | 1 |
| 81 | 55 | 47 | 46 | 1 |
| 82 | 55 | 56 | 47 | 1 |
| 83 | 56 | 48 | 47 | 1 |
| 84 | 56 | 57 | 48 | 1 |
| 85 | 57 | 49 | 48 | 1 |
| 86 | 57 | 58 | 49 | 1 |
| 87 | 58 | 50 | 49 | 1 |
| 88 | 58 | 59 | 50 | 1 |
| 89 | 59 | 51 | 50 | 1 |
| 90 | 59 | 60 | 51 | 1 |
| 91 | 60 | 52 | 51 | 1 |
| 92 | 60 | 61 | 52 | 1 |
| 93 | 61 | 53 | 52 | 1 |
| 94 | 61 | 62 | 53 | 1 |
| 95 | 62 | 54 | 53 | 1 |
| 96 | 62 | 63 | 54 | 1 |
| 97 | 64 | 56 | 55 | 1 |
| 98 | 64 | 65 | 56 | 1 |
| 99 | 65 | 57 | 56 | 1 |
| 100 | 65 | 66 | 57 | 1 |
| 101 | 66 | 58 | 57 | 1 |
| 102 | 66 | 67 | 58 | 1 |
| 103 | 67 | 59 | 58 | 1 |
| 104 | 67 | 68 | 59 | 1 |
| 105 | 68 | 60 | 59 | 1 |
| 106 | 68 | 69 | 60 | 1 |
| 107 | 69 | 61 | 60 | 1 |
| 108 | 69 | 70 | 61 | 1 |
| 109 | 70 | 62 | 61 | 1 |
| 110 | 70 | 71 | 62 | 1 |
| 111 | 71 | 63 | 62 | 1 |
| 112 | 71 | 72 | 63 | 1 |
| 113 | 64 | 73 | 65 | 1 |
| 114 | 73 | 66 | 65 | 1 |
| 115 | 66 | 77 | 67 | 1 |
| 116 | 77 | 68 | 67 | 1 |
| 117 | 68 | 78 | 69 | 1 |
| 118 | 78 | 70 | 69 | 1 |
| 119 | 70 | 82 | 71 | 1 |
| 120 | 82 | 72 | 71 | 1 |
| 121 | 73 | 74 | 66 | 1 |
| 122 | 74 | 75 | 66 | 1 |
| 123 | 75 | 76 | 66 | 1 |
| 124 | 76 | 77 | 66 | 1 |
| 125 | 78 | 79 | 70 | 1 |
| 126 | 79 | 80 | 70 | 1 |
| 127 | 80 | 81 | 70 | 1 |
| 128 | 81 | 82 | 70 | 1 |
| 129 | 83 | 84 | 64 | 1 |
| 130 | 84 | 73 | 64 | 1 |
| 131 | 84 | 74 | 73 | 1 |
| 132 | 84 | 85 | 74 | 1 |
| 133 | 85 | 75 | 74 | 1 |
| 134 | 85 | 86 | 75 | 1 |
| 135 | 86 | 87 | 75 | 1 |
| 136 | 75 | 87 | 76 | 1 |
| 137 | 87 | 88 | 76 | 1 |
| 138 | 76 | 88 | 77 | 1 |
| 139 | 86 | 89 | 77 | 1 |
| 140 | 77 | 89 | 68 | 1 |
| 141 | 80 | 90 | 68 | 1 |
| 142 | 68 | 90 | 78 | 1 |
| 143 | 90 | 91 | 78 | 1 |
| 144 | 78 | 91 | 79 | 1 |
| 145 | 91 | 92 | 79 | 1 |
| 146 | 79 | 92 | 80 | 1 |
| 147 | 93 | 94 | 83 | 1 |
| 148 | 94 | 84 | 83 | 1 |
| 149 | 94 | 95 | 84 | 1 |
| 150 | 95 | 85 | 84 | 1 |
| 151 | 95 | 96 | 85 | 1 |
| 152 | 96 | 86 | 85 | 1 |
| 153 | 96 | 97 | 86 | 1 |
| 154 | 97 | 98 | 86 | 1 |
| 155 | 86 | 98 | 87 | 1 |
| 156 | 98 | 99 | 87 | 1 |
| 157 | 87 | 99 | 88 | 1 |
| 158 | 99 | 100 | 88 | 1 |
| 159 | 88 | 100 | 89 | 1 |
| 160 | 100 | 101 | 89 | 1 |
| 161 | 89 | 101 | 102 | 1 |
| 162 | 89 | 102 | 90 | 1 |
| 163 | 90 | 102 | 103 | 1 |
| 164 | 90 | 103 | 91 | 1 |
| 165 | 91 | 103 | 104 | 1 |
| 166 | 91 | 104 | 92 | 1 |

**Table B2 (cont'd).**

**b. File NPD.**

| | | |
|---|---|---|
| 1 | 0.0000 | -0.1000 |
| 2 | 0.0383 | -0.0924 |
| 3 | 0.0707 | -0.0707 |
| 4 | 0.0924 | -0.0383 |
| 5 | 0.1000 | 0.0000 |
| 6 | 0.0924 | 0.0383 |
| 7 | 0.0707 | 0.0707 |
| 8 | 0.0383 | 0.0924 |
| 9 | 0.0000 | 0.1000 |
| 10 | 0.0000 | -0.1500 |
| 11 | 0.0574 | -0.1386 |
| 12 | 0.1061 | -0.1061 |
| 13 | 0.1386 | -0.0574 |
| 14 | 0.1500 | 0.0000 |
| 15 | 0.1386 | 0.0574 |
| 16 | 0.1061 | 0.1061 |
| 17 | 0.0574 | 0.1386 |
| 18 | 0.0000 | 0.1500 |
| 19 | 0.0000 | -0.2000 |
| 20 | 0.0765 | -0.1848 |
| 21 | 0.1414 | -0.1414 |
| 22 | 0.1848 | -0.0765 |
| 23 | 0.2000 | 0.0000 |
| 24 | 0.1848 | 0.0765 |
| 25 | 0.1414 | 0.1414 |
| 26 | 0.0765 | 0.1848 |
| 27 | 0.0000 | 0.2000 |
| 28 | 0.0000 | -0.3000 |
| 29 | 0.1148 | -0.2772 |
| 30 | 0.2121 | -0.2121 |
| 31 | 0.2772 | -0.1148 |
| 32 | 0.3000 | 0.0000 |
| 33 | 0.2772 | 0.1148 |
| 34 | 0.2121 | 0.2121 |
| 35 | 0.1148 | 0.2772 |
| 36 | 0.0000 | 0.3000 |
| 37 | 0.0000 | -0.4200 |
| 38 | 0.1607 | -0.3880 |
| 39 | 0.2970 | -0.2970 |
| 40 | 0.3880 | -0.1607 |
| 41 | 0.4200 | 0.0000 |
| 42 | 0.3880 | 0.1607 |
| 43 | 0.2970 | 0.2970 |
| 44 | 0.1607 | 0.3880 |
| 45 | 0.0000 | 0.4200 |
| 46 | 0.0000 | -0.5800 |
| 47 | 0.2220 | -0.5359 |
| 48 | 0.4101 | -0.4101 |
| 49 | 0.5359 | -0.2220 |
| 50 | 0.5800 | 0.0000 |
| 51 | 0.5359 | 0.2220 |
| 52 | 0.4101 | 0.4101 |
| 53 | 0.2220 | 0.5359 |
| 54 | 0.0000 | 0.5800 |
| 55 | 0.0000 | -0.7800 |
| 56 | 0.2985 | -0.7206 |
| 57 | 0.5515 | -0.5515 |
| 58 | 0.7206 | -0.2985 |
| 59 | 0.7800 | 0.0000 |
| 60 | 0.7206 | 0.2985 |
| 61 | 0.5515 | 0.5515 |
| 62 | 0.2985 | 0.7206 |
| 63 | 0.0000 | 0.7800 |
| 64 | 0.0000 | -1.0000 |
| 65 | 0.3827 | -0.9239 |
| 66 | 0.7071 | -0.7071 |
| 67 | 0.9239 | -0.3827 |
| 68 | 1.0000 | 0.0000 |
| 69 | 0.9239 | 0.3827 |
| 70 | 0.7071 | 0.7071 |
| 71 | 0.3827 | 0.9239 |
| 72 | 0.0000 | 1.0000 |
| 73 | 0.4142 | -1.0000 |
| 74 | 0.7071 | -1.0000 |
| 75 | 1.0000 | -1.0000 |
| 76 | 1.0000 | -0.7071 |
| 77 | 1.0000 | -0.4142 |
| 78 | 1.0000 | 0.4142 |
| 79 | 1.0000 | 0.7071 |
| 80 | 1.0000 | 1.0000 |
| 81 | 0.7071 | 1.0000 |
| 82 | 0.4142 | 1.0000 |
| 83 | 0.0000 | -1.4000 |
| 84 | 0.4000 | -1.4000 |
| 85 | 0.8000 | -1.4000 |
| 86 | 1.4000 | -1.4000 |
| 87 | 1.4000 | -1.0000 |
| 88 | 1.4000 | -0.6000 |
| 89 | 1.4000 | -0.2000 |
| 90 | 1.4000 | 0.2000 |
| 91 | 1.4000 | 0.6000 |
| 92 | 1.4000 | 1.0000 |
| 93 | 0.0000 | -2.0000 |
| 94 | 0.4000 | -2.0000 |
| 95 | 0.8000 | -2.0000 |
| 96 | 1.4000 | -2.0000 |
| 97 | 2.0000 | -2.0000 |
| 98 | 2.0000 | -1.4000 |
| 99 | 2.0000 | -1.0000 |
| 100 | 2.0000 | -0.6000 |
| 101 | 2.0000 | -0.2000 |
| 102 | 2.0000 | 0.2000 |
| 103 | 2.0000 | 0.6000 |
| 104 | 2.0000 | 1.0000 |

c. File BCT.

| | | |
|---|---|---|
| 1 | 1 | 100.0000 |
| 2 | 2 | 100.0000 |
| 3 | 3 | 100.0000 |
| 4 | 4 | 100.0000 |
| 5 | 5 | 100.0000 |
| 6 | 6 | 100.0000 |
| 7 | 7 | 100.0000 |
| 8 | 8 | 100.0000 |
| 9 | 9 | 100.0000 |
| 10 | 80 | 10.0000 |
| 11 | 81 | 10.0000 |
| 12 | 82 | 10.0000 |
| 13 | 72 | 10.0000 |
| 14 | 93 | 10.0000 |
| 15 | 94 | 10.0000 |
| 16 | 95 | 10.0000 |
| 17 | 96 | 10.0000 |
| 18 | 97 | 10.0000 |
| 19 | 98 | 10.0000 |
| 20 | 99 | 10.0000 |
| 21 | 100 | 10.0000 |
| 22 | 101 | 10.0000 |
| 23 | 102 | 10.0000 |
| 24 | 103 | 10.0000 |
| 25 | 104 | 10.0000 |
| 26 | 92 | 10.0000 |

d. File QUAN.

| | | | | | | |
|---|---|---|---|---|---|---|
| 104 | 166 | 26 | 0 | 0 | 0 | 6 |

e. File TIOT.

10.00
30.00
50.00
70.00
90.00
100.00

A facsimile catalog card in Library of Congress MARC format is reproduced below.


Albert, M.R.

Computer models for two-dimensional steady-state heat conduction / by M.R. Albert and G. Phetteplace. Hanover, N.H.: Cold Regions Research and Engineering Laboratory. Springfield, Va.: available from National Technical Information Service, 1983.

iv, 90 p., illus.; 28 cm. (CRREL Report 83-10.)

Prepared for Office of the Chief of Engineers by Corps of Engineers, U.S. Army Cold Regions Research and Engineering Laboratory under DA Project 4A762730 AT42.

Bibliography: p. 39.

1. Computer program documentation. 2. Computer programs. 3. Heat conduction. 4. Heat transfer. 5. Mathematical models. I. Phetteplace, G. II. United States. Army. Corps of Engineers. III. Cold Regions Research and Engineering Laboratory, Hanover, N.H. Series: CRREL Report 83-10.