

CRREL

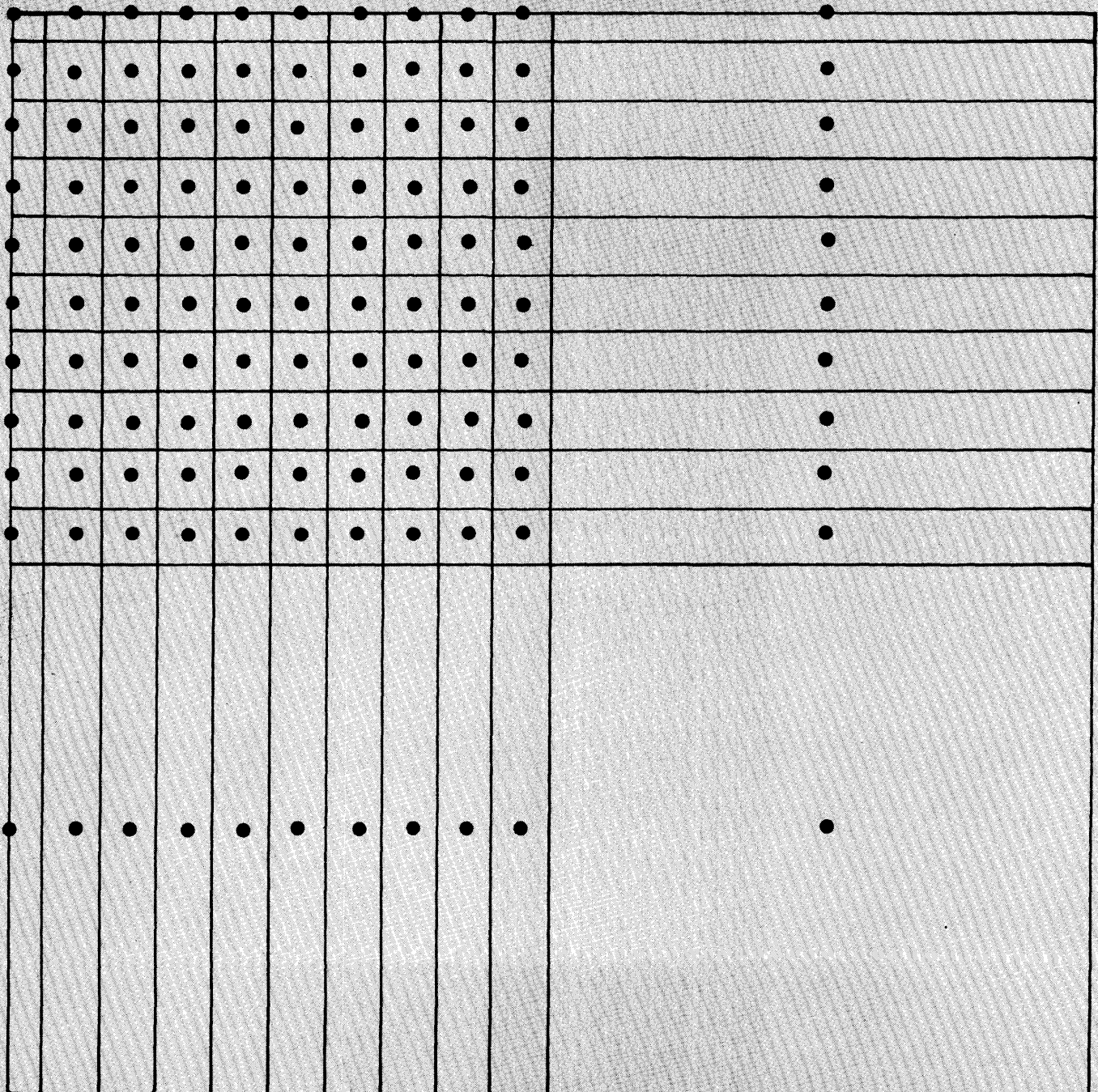
REPORT 83-12



US Army Corps
of Engineers

Cold Regions Research &
Engineering Laboratory

Computer models for two-dimensional transient heat conduction





CRREL Report 83-12

April 1983

Computer models for two-dimensional transient heat conduction

Mary Remley Albert

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CRREL Report 83-12	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER MODELS FOR TWO-DIMENSIONAL TRANSIENT HEAT CONDUCTION	5. TYPE OF REPORT & PERIOD COVERED	
	6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s) Mary Remley Albert	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 74	
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) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Centralized heating systems Computerized simulation Computer models Heat pipes Heat transfer		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper documents the development and verification of two finite difference models that solve the general two-dimensional form of the heat conduction equation, using the alternative-direction implicit method. Both can handle convective, constant flux, specified temperature and semi-infinite boundaries. The conducting medium may be composed of many materials. The first program, ADI, solves for the case where no change of state occurs. ADIPC solves for the case where a freeze/thaw change of phase may occur, using the apparent heat capacity method. Both models are verified by comparison to analytical results.		

PREFACE

This report was prepared by Mary Remley Albert, Mathematician, 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 author thanks Gary Phetteplace of CRREL for his support and technical review. Also, the author thanks Dr. Devinder Sodhi and Dr. Kevin O'Neill, both of CRREL, for their technical review.

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

	Page
Abstract	i
Preface	ii
Introduction	1
Finite differences applied to heat transfer	2
Heat conduction equation	2
Boundary conditions	6
Phase change	14
Computer program	16
ADDATA, the data subroutine	16
TRIDIG, the matrix solver	16
ISOTHM, the isotherm finder	17
ADI, main program	17
ADIPC, main program	18
Verification of ADI	18
Comparison of ADI with analytical results	18
Comparison of ADI with experimental results.....	23
Verification of ADIPC	25
Comparison of ADIPC with analytical results—the Neumann solution	25
Comparison of ADIPC with analytical results—two-dimensional phase change verification	27
User instruction for ADI	29
User instruction for ADIPC	29
Conclusions	30
Literature cited	30
Appendix A: Program INFSUM and sample input and output for program ADI	31
Appendix B. Program ADIPC and sample input and output.....	37

ILLUSTRATIONS

Figure

1. A simple finite difference grid	2
2. A node on the interior of the grid	3
3. A node on a right-hand boundary	6
4. A node on a semi-infinite boundary	8
5. An interior node adjacent to the semi-infinite boundary	9
6. A node on a corner of the grid	10
7. A semi-infinite corner	11
8. A semi-infinite side node adjacent to a semi-infinite corner	12
9. A node on a semi-infinite corner	13
10. Tridiagonal matrix equation	17
11. Semi-infinite corner problem	18
12. Finite difference grid for the semi-infinite corner problem	19
13. Locations of 35°, 30° and 25°C isotherms from ADI and from the analytical solution for the semi-infinite corner problem	19
14. Finite rectangle problem	20

Figure		Page
15.	Locations of 24°, 28°, 32°, 36° and 40°C isotherms from ADI and analytical solutions	21
16.	One-dimensional semi-infinite problem	22
17.	Thermocouple locations in the experimental setup	23
18.	Surface temperatures from experiment used in ADI runs	24
19.	Results of two ADI runs using different conductivities plotted against experimental results	25
20.	Comparison of ADIPC with Neumann solution	27
21.	Comparison of ADIPC with radial analytic freezing solution	28

TABLES

Table

1.	Comparison of ADI results to analytical results for the one-dimensional semi-infinite problem	22
----	---	----

COMPUTER MODELS FOR TWO-DIMENSIONAL TRANSIENT HEAT CONDUCTION

Mary Remley Albert

INTRODUCTION

Most major Army installations are heated with central heat distribution systems. The thermal regimes around buried distribution systems are of interest for several reasons, not the least of which is the estimated millions of dollars lost annually from these systems (Phetteplace et al. 1981). Also, when replacing damaged insulation or installing insulation in a new system, it is desirable to know what an optimum balance is between the initial cost of insulation and the continued operating cost of heat losses. In addition, freezing and thawing of the ground around a buried distribution system could damage it, through loss of support and settlement, for example.

Analysis of the thermal regime involves the solution of the heat conduction equation for situations with complicated geometries and a variety of boundary conditions. Usually, these partial differential equations cannot be solved analytically; we must resort to numerical methods. One long-established method is that of finite differences, a relatively straightforward numerical method used successfully to solve a variety of differential equations. A finite difference computer program, set up in a general form to solve the heat conduction equation under a variety of geometries and boundary conditions, provides a powerful tool with which the engineer can assess problems in conductive heat transfer.

The objective of this paper is to document the development and verification of two general two-dimensional finite difference computer programs that were written to model time-varying heat conduction in a medium composed of many materials. The first program, ADI, solves for the general case in which no change of state occurs in the conducting medium. The second program, ADIPC, is an adaptation of ADI that includes the effects of phase change in the conducting medium. The programs are able to handle convective, constant flux, specified temperature and semi-infinite boundaries. Material properties such as thermal conductivity, density and specific heat are allowed to vary with time or temperature. The programs are relatively easy to use, that is, they are easily set up for new conduction problems by those who have a background knowledge of FORTRAN.

The computer programs were written and programmed in FORTRAN by the author on CRREL's PRIME 400 computer.

Approved for release by NSA on 11-15-2013

FINITE DIFFERENCES APPLIED TO HEAT TRANSFER

Finite differences are commonly used in the numerical solution of partial differential equations. The procedure involves the replacement of differentials by differences, and is best illustrated by construction of the so-called finite difference grid. Let us examine a grid that is set up in two dimensions to model heat conduction (Fig. 1). The two dimensions in this case represent spatial independent variables, x and y . Each point of the grid is called a node and represents the area enclosed by the square around it (with unit depth). $T(x, y)$ represents the temperature of node x, y . For each node, the temperature and material properties are assumed uniform for the region it represents. By specifying the initial temperatures at each of the nodes, the nodal thermal properties and the boundary conditions, we can solve the heat conduction equation to determine the temperatures of the nodes at later times.

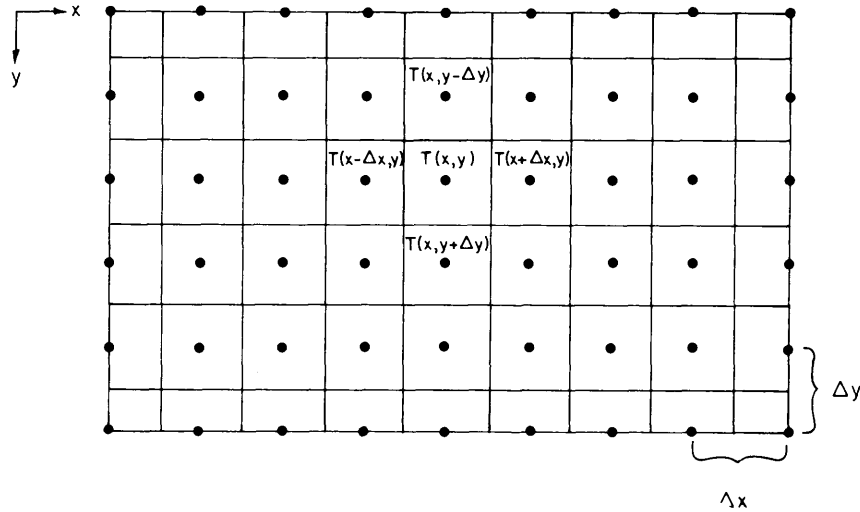


Figure 1. A simple finite difference grid.

Heat conduction equation

The equation governing transient heat conduction in two dimensions is

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) = \rho C_p \frac{\partial T}{\partial t} \quad (1)$$

where x and y = spatial variables

T = temperature

t = time

k = thermal conductivity (this may vary over the spatial and time domains)

ρ = density

C_p = specific heat.

The nonlinearities in this partial differential equation that arise from the inclusion of the phase change condition will be discussed in the *Phase Change* section.

Partial differential equations may be expressed in finite difference form either by a Taylor's series expansion about a point or by physical considerations, such as a heat flow balance in the case of the heat conduction problem. For problems involving variable thermal properties or complicated boundary conditions, the heat balance approach is the simplest.

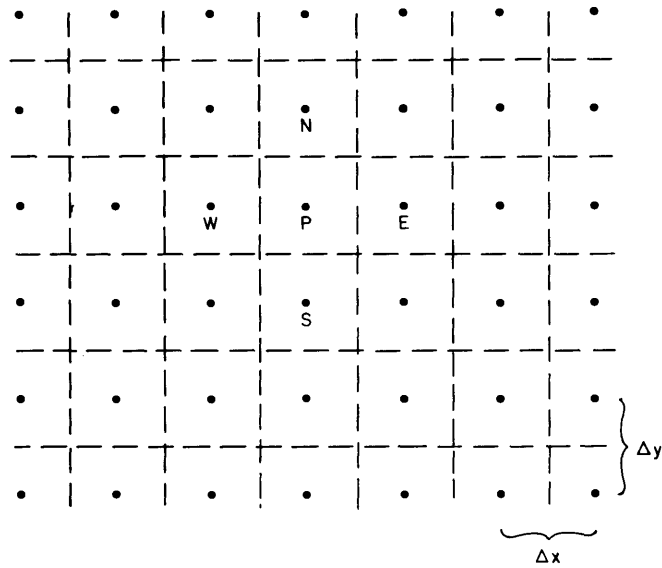


Figure 2. A node on the interior of the grid.

Consider the control volume represented by a node on the interior of the grid, illustrated in Figure 2. The heat flow Q into the control volume from any adjacent node may be calculated as follows:

$$Q = \frac{k}{L} A \Delta T \quad (2)$$

where k = thermal conductivity
 L = distance over which the heat flow occurs
 A = area (per unit depth) perpendicular to the direction of heat flow
 ΔT = change in temperature between the two nodes.

Examine the heat flow from node N to node P. The conductance k/L is the reciprocal of the resistance between the two nodes. The resistance between the two nodes R_{NP} is simply the sum of the individual resistances,

$$R_{NP} = R_N + R_P .$$

Each resistance is the distance divided by the material conductivity,

$$R_{NP} = \frac{\Delta y/2}{k_N} + \frac{\Delta y/2}{k_P}$$

then

$$\frac{k}{L} = \frac{1}{R_{NP}} = \frac{2 k_N k_P}{\Delta y (k_N + k_P)} . \quad (3)$$

In the computer program it is assumed that, for nodes not on a boundary of the grid, Δy and Δx are uniform and equal throughout the grid. Let the nodal spacing $\Delta x = \Delta y = \Delta s$. Following the form of eq 2, we find that the heat flows into the region associated with node P from nodes N, W, S and E may be calculated as follows:

$$Q_{NP} = \frac{2 k_N k_P}{\Delta s(k_N + k_P)} (\Delta s) (T_N - T_P) \quad (4)$$

$$Q_{WP} = \frac{2 k_W k_P}{\Delta s(k_W + k_P)} (\Delta s) (T_W - T_P) \quad (5)$$

$$Q_{SP} = \frac{2 k_S k_P}{\Delta s(k_S + k_P)} (\Delta s) (T_S - T_P) \quad (6)$$

$$Q_{EP} = \frac{2 k_E k_P}{\Delta s(k_E + k_P)} (\Delta s) (T_E - T_P) \quad (7)$$

Note that the area perpendicular to the direction of heat flow is taken per unit depth, that is, $A = \Delta s \cdot 1$.

The sum of the heat flows into a node is responsible for the temperature change of the node:

$$Q_{NP} + Q_{WP} + Q_{SP} + Q_{EP} = V_P \cdot \frac{1}{\Delta t} (\rho' C_P' T_P' - \rho C_P T_P) \quad (8)$$

where V_P is the control volume for unit depth ($V_P = (\Delta s)^2$ in this case) and the primed variables represent time $t + \Delta t$.

Now substitute eq 4-7 into eq 8 to get the finite difference form of eq 1 for a node not on the boundary of the grid:

$$k_{NP}(T_N - T_P) + k_{WP}(T_W - T_P) + k_{SP}(T_S - T_P) + k_E(T_E - T_P) = \frac{(\Delta s)^2}{\Delta t} (\rho' C_P' T_P' - \rho C_P T_P). \quad (9)$$

Since Δs is uniform for interior nodes of the grid, substitutions like the following have been made for simplicity's sake:

$$k_{NP} = \frac{2 k_N k_P}{k_N + k_P}. \quad (10)$$

Equation 8 describes the temperature change for one node P over one time step. The values ρ , C_P and k are held constant for each time step, but may be changed between steps. It is necessary to apply eq 8 (or the appropriate boundary equation) to each node of the grid to solve for the temperature distribution for one time step. The process is repeated until the desired number of time steps have been completed. Let us now investigate several methods used in the solution of the set of equations.

The first and probably the easiest method to program is the explicit method, where eq 8 is solved for T_P' (which is the value of T_P for time $t + \Delta t$). Note that all of the other temperature variables represent time t , so that the equation may be solved explicitly. An initial temperature is assigned to each node in the grid; the appropriate equation is solved for each node to determine the temperature resulting from the first time step. The resultant temperature distribution is then used in calculations for the second time step. This process continues until the desired number of time steps have been calculated. The main problem with this method is that a stability criterion must be met to produce a reliable answer. For a two-dimensional homogeneous grid, the criterion is

$$\frac{(\Delta s)^2 \rho C_P}{k \Delta t} > 4.$$

Note that a balance must be struck between the internodal distance (Δs) and the time step (Δt). In problems involving steep temperature gradients, a small internodal spacing must be used; consequently, the time step must be very small. The combination of many nodes and time steps makes the solution costly in computer time.

A second method used in solving finite difference equations is the implicit method. Here T'_p is taken as the value of T_p for time t , and all of the other variables represent time $t + \Delta t$; hence, all temperature variables are unknown except T'_p . An appropriate equation must be written for each node in the grid, and the resulting set of equations is solved simultaneously for each time step. For the two-dimensional case of an x by y grid, there will be xy equations in xy unknowns. The matrix of coefficients is banded (bandwidth = $2x + 1$), but in general there is no symmetry about the main diagonal. Large matrices often occur and they are usually solved by an iterative method. The implicit method has the advantage of being unconditionally stable, but it may require a fair number of iterations for adequate convergence in the matrix solution.

The finite difference method chosen for this computer program is known as the alternating direction implicit method (Peaceman and Rachford 1955, Carnahan et al. 1969). It is unconditionally stable like the fully implicit method, yet it requires only the solution of a tridiagonal matrix, thus being efficient in computer time and computer core storage. (The matrix-solving algorithm is discussed in the *TRIDIG, The Matrix Solver* section). Essentially, the method requires the solution of two different equations for each time step. The first equation is implicit only in the horizontal direction. The results from the first step are then used to solve the second equation, which is implicit only in the vertical direction.

For example, apply the alternating direction considerations to a node in the interior of the grid. Equation 9, which describes the heat balance for an interior node, is repeated here for convenience:

$$k_{NP}(T_N - T_P) + k_{WP}(T_W - T_P) + k_{SP}(T_S - T_P) + k_{EP}(T_E - T_P) = \frac{(\Delta s)^2 \rho' C'_P}{\Delta t} T'_P - \frac{(\Delta s)^2 \rho C_P}{\Delta t} T_P.$$

Each node in the grid is given an initial temperature. Then, for the first pass, let the above equation be implicit in the horizontal direction. The time step for the pass will be $\Delta t/2$, and the above equation may be written as follows:

$$k_{WP}T'_W + k_{EP}T'_E - \left[k_{WP} + k_{EP} + \frac{2(\Delta s)^2 \rho' C'_P}{\Delta t} \right] T'_P = -k_{NP}T_N - k_{SP}T_S + \left[k_{NP} + k_{SP} - \frac{2(\Delta s)^2 \rho C_P}{\Delta t} \right] T_P. \quad (11)$$

The primed variables represent the value of those variables at time $t + \Delta t/2$. The right-hand side of the equation is known; T'_W , T'_P and T'_E are to be determined. Equation 11 (or the appropriate boundary equation) is applied to each node of a row of the grid, creating a tridiagonal matrix of coefficients. Such a system of equations is quickly solved for each row of the grid. The resulting temperature distribution represents $t + \Delta t/2$, the end of the first pass.

The process is similar for the second pass, except that now eq 9 is rewritten to be implicit in the vertical direction:

$$k_{NP}T'_N + k_{SP}T'_S - \left[k_{NP} + k_{SP} + \frac{2(\Delta s)^2 \rho' C'_P}{\Delta t} \right] T'_P = -k_{WP}T'_W - k_{EP}T'_E + \left[k_{WP} + k_{EP} - \frac{2(\Delta s)^2 \rho C_P}{\Delta t} \right] T_P. \quad (12)$$

The primed variables represent time $t + \Delta t$, and the others represent time $t + \Delta t/2$. The tridiagonal matrix is formed and solved for each column of the grid; the resultant temperatures represent the temperature distribution in the grid after $t + \Delta t$, one time step. The entire process is repeated until the desired number of steps have been calculated.

The reader may consult Carnahan et al. (1969), Holman (1972), Croft and Lilley (1977) and Mitchell and Griffiths (1980) for more information on finite difference methods.

Boundary conditions

The following boundary conditions will be derived from heat balance considerations. In this paper each boundary equation will not be expressed in the form needed for use in the alternating direction procedure, but the reader should be aware that each equation was put into that form for use in the computer programs ADI and ADIPC.

Sides of the grid

Consider a node, P, on the right-hand grid boundary. The control volume associated with the node is the area enclosed by dotted lines around it, as depicted in Figure 3.

The sum of the heat flowing through the boundaries of P's control volume is responsible for the temperature change of the node, that is,

$$Q_{NP} + Q_{WP} + Q_{SP} + Q_{EP} = \frac{\Delta x \Delta y \rho C_p \Delta T_p}{2\Delta t} \quad (13)$$

where Q_{NP} = the heat flow from node N to (or from) node P

$\frac{\Delta x}{2} \cdot \Delta y$ = the volume (for unit depth) of node P

ρ = the density of the material in node P

C_p = the specific heat

T = temperature

t = time.

The flow of heat from node N to node P is given by

$$Q_{NP} = \frac{1}{2} k_{NP} (T_N - T_p) \quad (14)$$

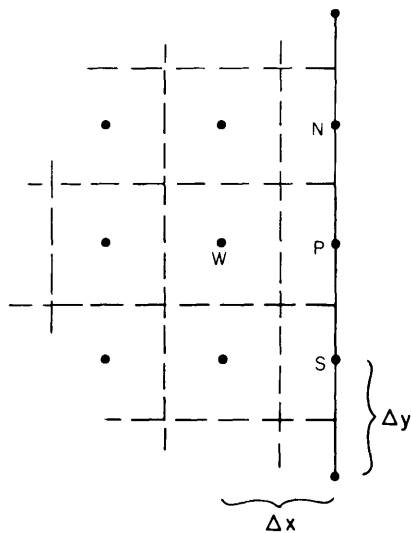


Figure 3. A node on a right-hand boundary.

where k_{NP} is the effective conductivity between nodes N and P, and for $\Delta x = \Delta y = \Delta s$,

$$k_{NP} = \frac{2k_N k_P}{(k_N + k_P)}$$

Similarly, the heat flow from nodes W and S to node P may be given as

$$Q_{WP} = k_{WP} (T_W - T_P) \quad (15)$$

$$Q_{SP} = \frac{1}{2} k_{SP} (T_S - T_P). \quad (16)$$

Now consider several cases describing the heat flow across the boundary.

Constant flux boundary. For a boundary subject to a constant heat flux, the condition

$$\left. \frac{\partial T}{\partial x} \right|_{x=P} = \text{constant}$$

exists at the boundary. For node P, illustrated in Figure 3, the heat flow across its east side is given by

$$Q_{EP} = \phi \cdot \Delta y \quad (17)$$

where ϕ is the heat flux per unit area crossing the boundary. The equation for a node on the right-hand side of the grid with constant flux is obtained by combining eq 13-17 and allowing $\Delta x = \Delta y = \Delta s$:

$$\frac{1}{2} k_{NP} (T_N - T_P) + k_{WP} (T_W - T_P) + \frac{1}{2} k_{SP} (T_S - T_P) + \phi \Delta s = \frac{(\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P]. \quad (18)$$

The indices may be suitably rearranged for constant flux boundaries on other sides of the grid.

Note that for a boundary which is insulated or on a line of symmetry the zero heat flux condition holds, and $\phi = 0$.

Convection boundary. For the node illustrated in Figure 3, exposed to convection on the right-hand side, the heat flow from the convective medium to node P is given by

$$Q_{EP} = h_E \Delta y (T_E - T_P) \quad (19)$$

where h_E is the coefficient of convective heat transfer and T_E is the temperature outside the grid. Combining equations 13, 14, 15, 16 and 19, we arrive at the equation for a node on the convective right-hand side boundary, for $\Delta x = \Delta y = \Delta s$:

$$\frac{1}{2} k_{NP} (T_N - T_P) + k_{WP} (T_W - T_P) + \frac{1}{2} k_{SP} (T_S - T_P) + h \Delta s (T_E - T_P) = \frac{(\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P]. \quad (20)$$

Specified temperature boundary. For a node of specified temperature on a boundary, corner or inside the grid, apply the equation $T_P = C$, where C is the temperature of the node at time t .

Semi-infinite boundary. This condition represents a continuous, uniform material extending in one direction, with a known temperature a large distance away. It is approximated here by use

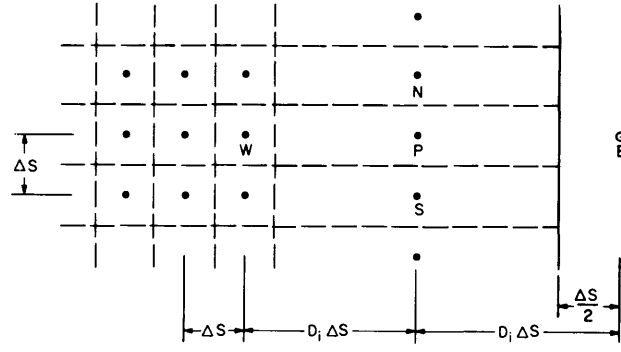


Figure 4. A node on a semi-infinite boundary.

of a large internodal distance between the last two nodes of a row of the grid. Consider the situation illustrated in Figure 4, for the right-hand side of the grid. Node E is not actually a part of the grid, but is the location of known temperature T_E outside the grid.

The distance between the last two nodes and also between the boundary node and the location of the known temperature T_E is $D_i \Delta s$. Note that D_i represents a multiple of Δs . The heat flow from nodes N, W, S and E to node P may be calculated as follows

$$Q_{NP} = k_{NP} (2D_i - 1) (T_N - T_P) \quad (21)$$

$$Q_{WP} = k_{WP} (T_W - T_P) \quad (22)$$

$$Q_{SP} = k_{SP} (2D_i - 1) (T_S - T_P) \quad (23)$$

$$Q_{EP} = k_{EP} (T_E - T_P) \quad (24)$$

The effective conductivities are of the same form as eq 10, except for k_{WP} and k_{EP}

$$k_{WP} = \frac{2k_W k_P}{[k_P + (2D_i - 1) k_W]} \quad .$$

Node E is assumed to be the same material as node P, thus $k_{EP} = k_P$. The sum of the heat flows accounts for the temperature change of the node, thus the equation for a node on the right-hand semi-infinite boundary is

$$k_{NP} (2D_i - 1) (T_N - T_P) + k_{WP} (T_W - T_P) + k_{SP} (2D_i - 1) (T_S - T_P) + k_{EP} (T_E - T_P) = \frac{(2D_i - 1) (\Delta s)^2}{\Delta t} (\rho' C'_P T'_P - \rho C_P T_P) \quad (25)$$

where $(2D_i - 1) (\Delta s)^2$ is the control volume for the node per unit depth.

In any finite difference formulation, the accuracy of the solution increases as the area represented by a node decreases. Therefore, when using the semi-infinite boundary formulation, the user should specify the smallest D_i acceptable. The semi-infinite condition should only be used in regions where the temperature gradient is small and precise knowledge of the temperature distribution is not critical.

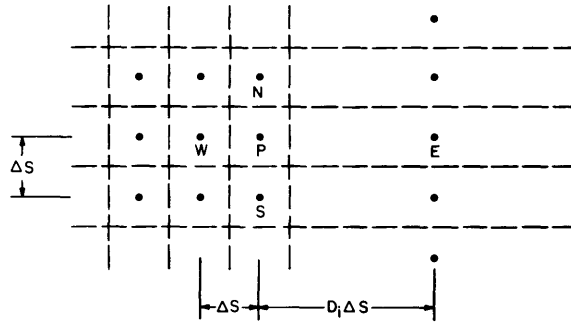


Figure 5. An interior node adjacent to the semi-infinite boundary.

This boundary conditions also requires a special heat balance for the node adjacent to the semi-infinite node. Consider again the situation for a right-hand boundary, as illustrated in Figure 5. The appropriate heat flow equations are

$$Q_{NP} = k_{NP} (T_N - T_P) \quad (26)$$

$$Q_{WP} = k_{WP} (T_W - T_P) \quad (27)$$

$$Q_{SP} = k_{SP} (T_S - T_P) \quad (28)$$

where k_{NP} , k_{WP} , k_{SP} are of the same form as eq 10, and

$$Q_{EP} = k_{EP} (T_N - T_P) \quad (29)$$

where $k_{EP} = \frac{2k_E k_P}{[k_E + (2D_i - 1) k_P]}$.

The equation for an interior node adjacent to a right-hand semi-infinite boundary node is

$$k_{NP} (T_N - T_P) + k_{WP} (T_W - T_P) + k_{SP} (T_S - T_P) + k_{EP} (T_E - T_P) = \frac{(\Delta s)^2}{\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \quad (30)$$

Corner of the grid

The corners of the grid require special heat balances, depending on the conditions on each edge of the grid. Consider the upper right-hand corner of the grid, shown in Figure 6. T_N and T_E are not a part of the grid, but are temperatures outside the grid. The heat flow from the two nodes adjacent to node P may be given as

$$Q_{WP} = \frac{1}{2} k_{WP} (T_W - T_P) \quad (31)$$

$$Q_{SP} = \frac{1}{2} k_{SP} (T_S - T_P) \quad (32)$$

where k_{WP} and k_{SP} follow the form given in eq 10. Q_{NP} and Q_{EP} are dependent upon the

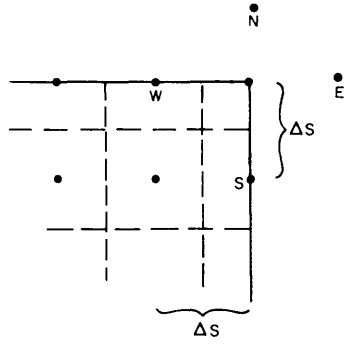


Figure 6. A node on a corner of the grid.

particular boundary conditions and will be presented shortly. The corner equations will follow the form

$$Q_{NP} + Q_{WP} + Q_{SP} + Q_{EP} = \frac{(\Delta s)^2}{4} \rho C_P \frac{\Delta T}{\Delta t} \quad (33)$$

where $\frac{(\Delta s)^2}{4}$ is the volume represented by node P for unit depth.

Constant flux on both sides. Let ϕ_N be the heat flux per unit area crossing the north side of the corner shown in Figure 6, and ϕ_E be that crossing the east side of the corner. Then the heat flows across the two sides are given by

$$Q_{NP} = \phi_N \cdot \frac{\Delta s}{2} \quad (34)$$

$$Q_{EP} = \phi_E \cdot \frac{\Delta s}{2} \quad (35)$$

The equation for the corner is

$$\frac{1}{2} k_{WP} (T_W - T_P) + \frac{1}{2} k_{SP} (T_S - T_P) + \frac{1}{2} \Delta s (\phi_N + \phi_E) = \frac{(\Delta s)^2}{4\Delta t} [\rho' C_P' T_P' - \rho C_P T_P] \quad (36)$$

As in the case of the side of the grid, if a side of the corner is insulated or on a line of symmetry, $\phi = 0$ for that side.

Convection on both sides. For a corner subject to convection on both of its sides, the heat flow from each of the two sides may be given as

$$Q_{NP} = h_N \frac{\Delta s}{2} (T_N - T_P) \quad (37)$$

$$Q_{EP} = h_E \frac{\Delta s}{2} (T_E - T_P) \quad (38)$$

where h_N is the coefficient of convective heat transfer on the north side and h_E is that on the east side. The equation for node P is

$$\frac{1}{2} k_{WP} (T_W - T_P) + \frac{1}{2} k_{SP} (T_S - T_P) + \frac{1}{2} h_N \Delta s (T_N - T_P) + \frac{1}{2} h_E \Delta s (T_E - T_P) = \frac{(\Delta s)^2}{4\Delta t} [\rho' C_P' T_P' - \rho C_P T_P] \quad (39)$$

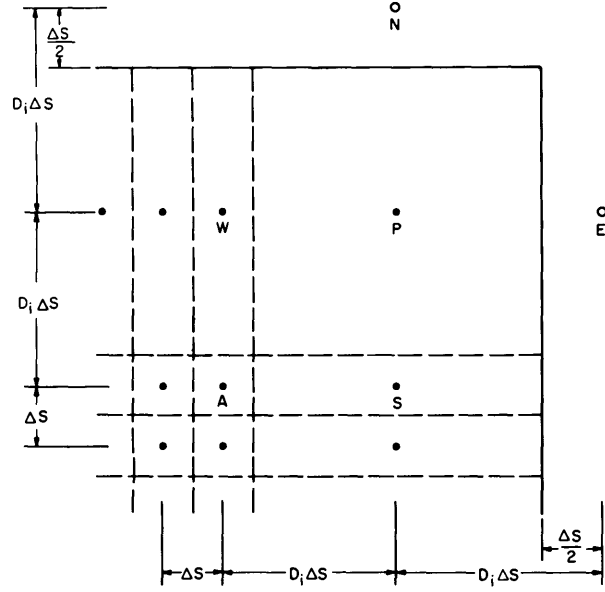


Figure 7. A semi-infinite corner.

Semi-infinite on both sides. For a corner with semi-infinite conditions on both sides, an irregular finite difference grid is employed once more. Consider again the upper right-hand corner, now illustrated in Figure 7. T_N and T_E are known temperatures outside the grid. Heat flows are given by

$$Q_{NP} = k_{NP} (2D_i - 1) (T_N - T_P) \quad (40)$$

$$Q_{WP} = k_{WP} (2D_i - 1) (T_W - T_P) \quad (41)$$

$$Q_{SP} = k_{SP} (2D_i - 1) (T_S - T_P) \quad (42)$$

$$Q_{EP} = k_{EP} (2D_i - 1) (T_E - T_P). \quad (43)$$

Again,

$$k_{NP} = k_{EP} = k_P; k_{WP} = \frac{2k_W k_P}{k_P + (2D_i - 1) k_W} \text{ and } k_{SP} = \frac{2k_S k_P}{k_P + (2D_i - 1) k_S}.$$

Then the equation for a semi-infinite corner node follows:

$$k_{NP} (2D_i - 1) (T_N - T_P) + k_{WP} (2D_i - 1) (T_W - T_P) + k_{SP} (2D_i - 1) (T_S - T_P) + k_{EP} (2D_i - 1) (T_E - T_P) = \frac{[(2D_i - 1) (\Delta s)]^2}{\Delta t} [\rho' C'_p T'_p - \rho C_p T_p]. \quad (44)$$

Because the nodes are of irregular size, special consideration must be given also to node temperatures T_W , T_S and T_A in Figure 7.

Consider first node A. This is similar to the node P illustrated in Figure 5, except that the distance to node N is now $D_i \cdot \Delta s$. This results in our changing k_{NP} to

$$k_{NP} = \frac{2k_N k_P}{k_N + (2D_i - 1) k_P}.$$

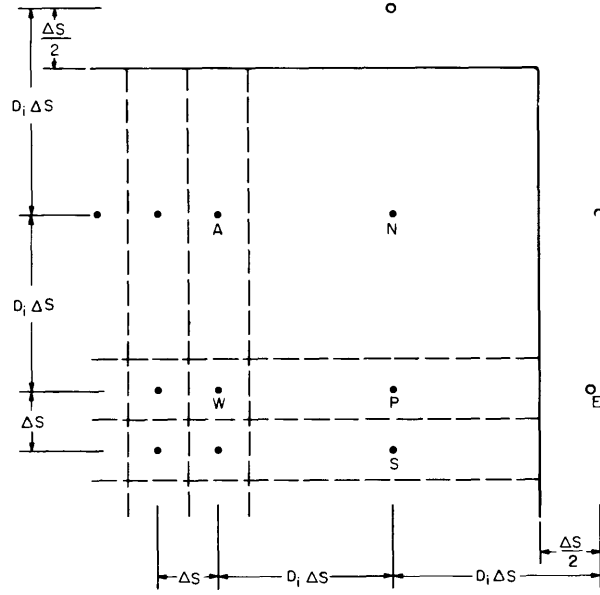


Figure 8. A semi-infinite side node adjacent to a semi-infinite corner.

Except for this change in k_{NP} , the equation for the square interior node inside an upper right-hand corner with semi-infinite conditions on both sides is the same as eq 29.

Now consider the semi-infinite nodes adjacent to the semi-infinite corner node. These are node temperatures T_W and T_S in Figure 7. The figure is presented again, this time as Figure 8, with the nodes relabeled. The type of node under consideration is represented by nodes A and P in Figure 8. The following equations reflect the heat flow to node P:

$$Q_{NP} = k_{NP} (2D_i - 1) (T_N - T_P) \quad (45)$$

$$Q_{WP} = k_{WP} (T_W - T_P) \quad (46)$$

$$Q_{SP} = k_{SP} (2D_i - 1) (T_S - T_P) \quad (47)$$

$$Q_{EP} = k_{EP} (T_S - T_P) . \quad (48)$$

Here,

$$k_{NP} = \frac{2k_N k_P}{k_N + (2D_i - 1) k_P} \quad \text{and} \quad k_{WP} = \frac{2k_W k_P}{k_P + (2D_i - 1) k_W} .$$

The equation for this node is

$$k_{NP} (2D_i - 1) (T_N - T_P) + k_{WP} (T_W - T_P) + k_{SP} (2D_i - 1) (T_S - T_P) + k_{EP} (T_E - T_P) = \frac{(2D_i - 1) (\Delta s)^2}{\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \quad (49)$$

Constant flux, convective corner. For the corner illustrated in Figure 6, allow convection to occur across the north side of node P and constant flux across the east side

$$Q_{NP} = h_N \frac{\Delta s}{2} (T_N - T_P) \quad (50)$$

Q_{EP} is given by eq 35, and Q_{WP} and Q_{SP} are given by eq 31 and 32, respectively. The resulting equation for the corner is:

$$\begin{aligned} \frac{1}{2} h_N \Delta s (T_N - T_P) + \frac{1}{2} k_{WP} (T_W - T_P) + \frac{1}{2} k_{SP} (T_S - T_P) + \frac{1}{2} \phi_E \Delta s = \\ \frac{(\Delta s)^2}{4\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \end{aligned} \quad (51)$$

Constant flux, semi-infinite corner. The corner illustrated in Figure 9 has a semi-infinite boundary on the right side. Let there be a constant flux per unit area, ϕ_N , across the north side of node P, then

$$Q_{NP} = \phi_N (2D_i - 1) \Delta s \quad (52)$$

$$Q_{WP} = \frac{1}{2} k_{WP} (T_W - T_P) \quad (53)$$

$$Q_{SP} = k_{SP} (2D_i - 1) (T_S - T_P) \quad (54)$$

$$Q_{EP} = \frac{1}{2} k_{EP} (T_E - T_P)$$

$$k_{WP} = \frac{2k_W k_P}{k_P + (2D_i - 1) k_W}$$

$$k_{SP} = \frac{2k_S k_P}{k_S + k_P}$$

$$k_{EP} = k_P . \quad (55)$$

The equation for the corner is as follows:

$$\begin{aligned} \phi_N (2D_i - 1) \Delta s + \frac{k_{WP}}{2} (T_W - T_P) + k_{SP} (2D_i - 1) (T_S - T_P) + \frac{k_{EP}}{2} (T_E - T_P) = \\ \frac{(2D_i - 1) (\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \end{aligned} \quad (56)$$

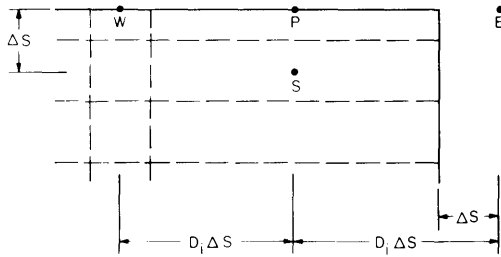


Figure 9. A node on a semi-infinite corner.

Again, the node adjacent to the semi-infinite node must have a special equation. For node W in Figure 9, with constant flux ϕ_N across the top, eq 57 applies, with reference to the node under consideration as node P:

$$\phi_N \Delta s + \frac{1}{2} k_{WP} (T_W - T_P) + k_{SP} (T_S - T_P) + \frac{1}{2} k_{EP} (T_E - T_P) = \frac{(\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \quad (57)$$

Semi-infinite, convective corner. For the corner in Figure 9, allow convection across the north side, and allow the same semi-infinite boundary on the east side,

$$Q_{NP} = h_N (2D_i - 1) (T_N - T_P) . \quad (58)$$

Q_{WP} , Q_{SP} and Q_{EP} appear in eq 53, 54 and 55, respectively. Thus the equation for the configuration is

$$h_N (2D_i - 1) (T_N - T_P) + \frac{k_{WP}}{2} (T_W - T_P) + k_{SP} (2D_i - 1) (T_S - T_P) + \frac{k_{EP}}{2} (T_E - T_P) = \frac{(2D_i - 1) (\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \quad (59)$$

The appropriate equation for node W in Figure 9, subject to convection across the top, follows. Again, refer to this node as node P:

$$h_N \Delta s (T_N - T_P) + \frac{1}{2} k_{WP} (T_W - T_P) + k_{SP} (T_S - T_P) + \frac{k_{WP}}{2} (T_W - T_P) = \frac{(\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \quad (60)$$

PHASE CHANGE

The methods discussed in this report so far apply to both ADIPC and ADI, the programs developed for heat conduction with and without phase change. The ideas on phase change presented in this section will apply only to ADIPC. The computer program solves the heat conduction equation only; possible effects of moisture migration through the medium (and unfrozen water content in a frozen soil, for example) are neglected. Although the examples used involve freezing, ADIPC permits either freezing or thawing at any node of the grid.

During the freezing process, the temperature of a substance that is initially above its fusion temperature decreases as heat is removed until the substance reaches its fusion temperature. The continuing removal of heat produces no change in temperature until the energy equivalent to the latent heat of the substance is removed. The substance has then changed state to become frozen; any further removal of heat again results in a decrease of temperature below the temperature of fusion.

ADIPC models freezing or thawing by defining an apparent specific heat during phase change that accounts for the entire enthalpy change that takes place, including the enthalpy in the latent heat of fusion. To do this, it must be assumed that phase change takes place over a finite temperature range, ΔT , around the fusion temperature. Allow T_f to denote the fusion temperature, C_{PA}

and C_{PB} to be the specific heat of the substance above and below freezing, respectively, and let C_P^* be the apparent specific heat defined for phase change. For a temperature change in a substance not involving a change of state, the enthalpy change ΔH is given as

$$\Delta H = \int C_P dT .$$

Now the temperature range for phase change will be $T_f \pm \Delta T/2$, and the apparent specific heat accounts for the entire enthalpy change as follows, where H_L represents the latent heat of fusion:

$$\int_{T_f - \frac{\Delta T}{2}}^{T_f + \frac{\Delta T}{2}} C_P^* dT = \int_{T_f - \frac{\Delta T}{2}}^{T_f} C_{PB} dT + \int_{T_f}^{T_f + \frac{\Delta T}{2}} C_{PA} dT + H_L . \quad (61)$$

ΔT is chosen small enough so that each specific heat may be assumed constant over the integral,

$$C_P^* \Delta T = C_{PB} \frac{\Delta T}{2} + C_{PA} \frac{\Delta T}{2} + H_L$$

$$C_P^* = \frac{1}{2} (C_{PB} + C_{PA}) + \frac{H_L}{\Delta T} . \quad (62)$$

Equation 62 defines the apparent heat capacity as used in ADIPC. The apparent heat capacity method is further discussed in Bonacina and Comini (1973).

Phase change is implemented in ADIPC as follows. At the beginning of each time step, each node of the grid is examined. If its temperature lies within the range $T_f \pm \Delta T/2$, the specific heat for that node is defined by eq 62. The conduction equations are set up and solved as usual. At the end of the complete time step, the newly calculated temperature of each node is compared with the temperature of that node at the beginning of the time step. If the temperature skipped from below $T_f - \Delta T/2$ to above $T_f + \Delta T/2$, or vice versa, then the phase change front skipped that node, and the temperature of the node is reassigned as follows. If the temperature of the node skipped from the frozen to the unfrozen domain,

$$T = \left(T_f - \frac{\Delta T}{2} \right) + \frac{C_{PB}}{C_P^*} \left[T' - \left(T_f - \frac{\Delta T}{2} \right) \right] \quad (63)$$

where T' is the calculated nodal temperature before its reassignment. If the node skipped from an unfrozen state to the frozen state, its temperature is reassigned as follows:

$$T = \left(T_f + \frac{\Delta T}{2} \right) + \frac{C_{PA}}{C_P^*} \left[T' - \left(T_f + \frac{\Delta T}{2} \right) \right] . \quad (64)$$

In this way, the program assures that the phase change front does not skip a node.

A drawback with the apparent heat capacity method is that it is not designed to follow the exact location of the phase change front for each time step; it is designed to calculate nodal temperature. The location of the T_f isotherm may be found by interpolation, but it approximates the location of the front in a step-like pattern because of the discretization of a continuous space, as will be illustrated in the program verifications. This behavior is minimized by use of a small internodal spacing. The method is flexible enough to deal with phase change in two-dimensional space when there may be several locations of phase change fronts; for this flexibility the method was chosen.

COMPUTER PROGRAM

In using finite differences, we replace differentials and derivatives with differences. It makes sense, then, that the accuracy of the solution increases as these differences become smaller. Specifically, the accuracy improves as the nodal spacing and time step, Δs and Δt , are made small. There is no set criterion for just how small they must be, usually this is discovered through trial and error.

The grid for these programs may be any size, so far as the outside dimensions are concerned, but, except for the semi-infinite boundary conditions, each node in the grid represents a square area of unit depth. RAY (I, J, K) is the array that represents the grid. I and J are spatial variables in Cartesian coordinates; I is incremented vertically and J horizontally. K has three values. RAY (I, J, 1) represents the temperature distribution in the grid, i.e., a temperature is stored at each I, J location. RAY (I, J, 2) records the nodal location type. Examples of location types include a node on a constant flux boundary, variable interior node, specified temperature corner node, etc. Each location type is assigned a number, and one such number is stored for each I, J location under RAY (I, J, 2). This information is used to assign the correct form of the heat conduction equation to each node. Each material type in the problem is given a number; these numbers are stored in RAY (I, J, 3) and may be used to assign a conductivity, density and specific heat for each node.

The programs are made up of four parts: 1) a data-gathering subroutine, 2) the main program, 3) a subroutine to solve the tridiagonal matrix and 4) a subroutine to locate the user-specified isotherms at times specified by the user. Parts 3 and 4 are identical for both ADI and ADIPC, and part 1 is similar for each. Therefore, parts 1, 3 and 4 will be discussed only once but are included in each of the two programs.

ADDATA, the data subroutine

The necessary data for the program are handled mainly through subroutine ADDATA. Variables used are defined in the comment statements at the beginning of the subroutine. The user has simply to edit the subroutine, following directions in the comment statements in the subroutine to initialize variables and arrays. When the subroutine is run, the data are put into a formatted data file ADIDAT; the user does not have to worry about formats.

TRIDIG, the matrix solver

This subroutine solves the tridiagonal matrix formed in the main program. The matrix is formed by the application of eq 11 or 12 (or a boundary condition counterpart) to each node of the grid. The resultant system of equations is illustrated in matrix form in Figure 10. The matrix of coefficients is an n by n matrix; all entries are zero except those on the three center diagonals, hence the term tridiagonal matrix. The vector on the right contains the elements of the right-hand side of each equation. To conserve computer storage space, only the three diagonals are stored, as vectors. Thus the effective storage space required is reduced from n by $n + 1$ to n by 4.

The solution algorithm is commonly used (e.g. Gerald 1980). Each element of the lower diagonal may be eliminated by subtracting the appropriate multiple of the $(i - 1)$ row from the i th row. The values of b_i and d_i , after elimination of a_i , are

$$b_i = b_i - \left(\frac{a_i}{b_{i-1}} \right) c_{i-1} \quad (65)$$

$$d_i = d_i - \left(\frac{a_i}{b_{i-1}} \right) d_{i-1} \quad \text{for } i = 1, 2, 3 \dots n . \quad (66)$$

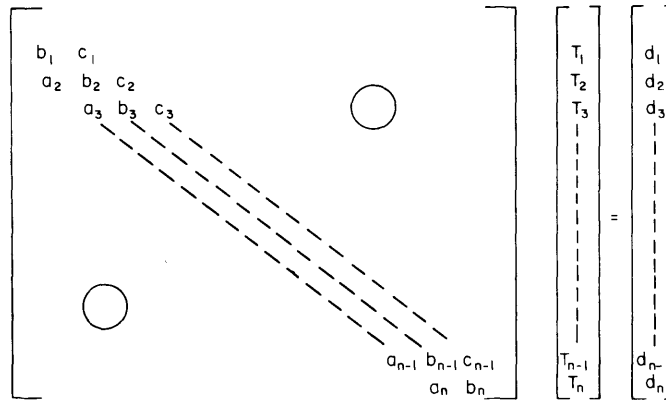


Figure 10. Tridiagonal matrix equation.

After replacing the element of vectors b and d by the new values, we perform a back substitution as follows:

$$d_n = \frac{d_n}{b_n} \quad (67)$$

$$d_i = \frac{d_i - c_i d_{i+1}}{b_i}, \quad \text{for } i = n - 1, n - 2, \dots, 1. \quad (68)$$

The elements of the solution vector replace vector d .

ISOTHM, the isotherm finder

Subroutine ISOTHM examines the temperature distribution in the grid and performs a linear interpolation between adjacent nodes to produce the Cartesian coordinates of the locations of the user-specified isotherms. The coordinates are listed in file POINTS in two columns, one for the horizontal coordinate and one for the vertical coordinate. POINTS may be used as a data file for a plotting routine. Subroutine ISOTHM may be called at the completion of any number of time steps; the frequency is specified by the user in ADDATA.

ADI, main program

ADI first interactively asks the user if ADDATA should be called. The subroutine should be called if ADIDAT does not already contain the formatted data. If the user so desires, ADDATA is called. If not, ADI reads the data from ADIDAT.

For each time step, the thermal properties such as conductivity, density and specific heat are updated, if appropriate, and the resultant conductivities between the nodes are figured. Each node is assigned the coefficients for the appropriate form of the conduction equation, forming the tridiagonal matrix (Fig. 10), and subroutine TRIDIG is called to solve the matrix. The resultant temperature distribution is then used in the second half-time step, and TRIDIG is called once again. These temperatures then represent the distribution for one complete time step. If appropriate, subroutine ISOTHM is then called to locate the user-specified isotherms. This procedure is repeated for the desired number of time steps.

The initial data and boundary conditions are printed into file ADIOUT; the temperature distributions for specified time steps are printed into file ADITMP, and finally, a new data file, ADNDAT, is created for the final temperature distribution. This file is in the same format as the input file ADIDAT; ADNDAT may be used as a starting point if the user wishes to run the model for more time.

ADIPC, main program

ADIPC is similar to ADI, but differs in that phase change considerations are implemented. At the beginning of each time step, if the temperature of a node lies in the phase change temperature range, the program assigns the apparent specific heat to that node, as previously discussed. Otherwise, the specific heat, thermal conductivity and density for each node are updated according to the user's specifications.

At the end of the time step, the temperature of each node is compared to its temperature at the end of the previous time step. If the temperature skipped over the phase change temperature range, it is reassigned as discussed in the *Phase Change* section. Then, if appropriate, subroutine ISOTHM is called.

The initial and boundary conditions are printed in readable form in file ADPOUT; the temperature distributions for specified time steps are printed into file ADPTMP, and a new data file with the final temperature distributions is printed into file ADPNDT.

VERIFICATION OF ADI

Comparison of ADI with analytical results

Semi-infinite corner

The results of ADI will first be compared to the problem of a semi-infinite corner, as illustrated in Figure 11.

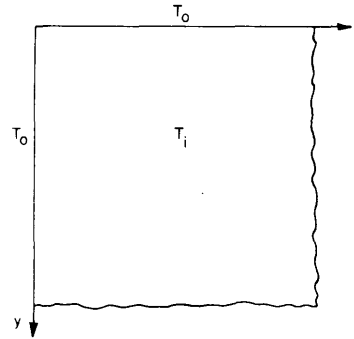


Figure 11. Semi-infinite corner problem.

T_i is the uniform initial temperature; at time $t = 0$ the temperature of the two edges, given by $x = 0$ and $y = 0$, is changed to T_0 . The solution is well documented (Carslaw and Jaeger 1959, and Holman 1972) and is found by using a product solution for two one-dimensional problems. The solution is given by

$$\frac{T - T_0}{T_i - T_0} = \operatorname{erf} \frac{x}{2\sqrt{\alpha t}} \operatorname{erf} \frac{y}{2\sqrt{\alpha t}} \quad (69)$$

where α is the thermal diffusivity, and

$$\operatorname{erf} \frac{x}{2\sqrt{\alpha t}} = \frac{2}{\sqrt{\pi}} \int_0^{x/(2\sqrt{\alpha t})} e^{-\eta^2} d\eta.$$

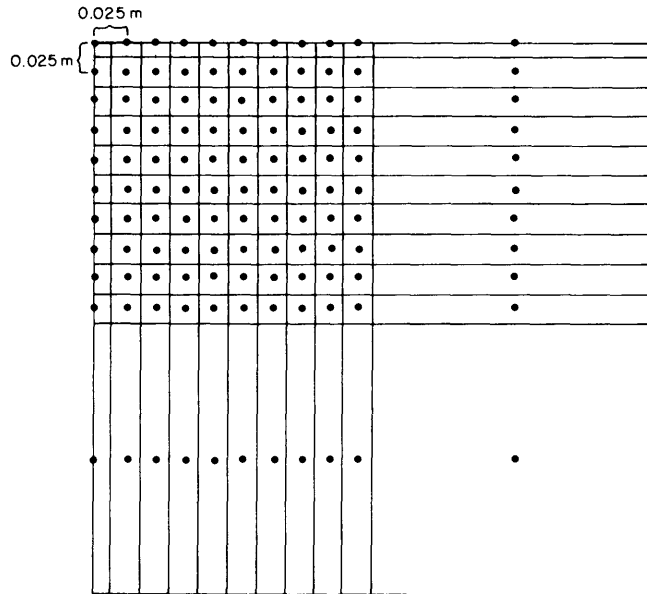


Figure 12. Finite difference grid for the semi-infinite corner problem.

For a comparison run, the following values were used:

$$T_i = 20^\circ\text{C} (68^\circ\text{F})$$

$$T_0 = 40^\circ\text{C} (104^\circ\text{F})$$

$$\alpha = 0.0025 \text{ m}^2/\text{hr}$$

The finite difference grid appears as shown in Figure 12. The boundary condition assigned to the top and left side was that of constant temperature equal to 40°C . The right side and bottom of the grid were assigned a semi-infinite boundary condition, with $T_S = T_E = 20^\circ\text{C}$ and $D_i = 50$ (see the *Sides of the Grid* section). The nodes not situated on a boundary were assigned an initial temperature of 20°C . A time step of 0.25 hr was used, and the internodal spacing was 0.025 m.

The locations of the 35° , 30° and 25°C isotherms are plotted for the results of ADI and for the analytical solution for several time steps in Figure 13. Excellent agreement is found between the two solutions for regions not adjacent to the semi-infinite boundary. As previously stated, the semi-infinite condition is an approximation. Also, the large internodal distance used for the semi-infinite boundary decreases the accuracy of the solution in that region.

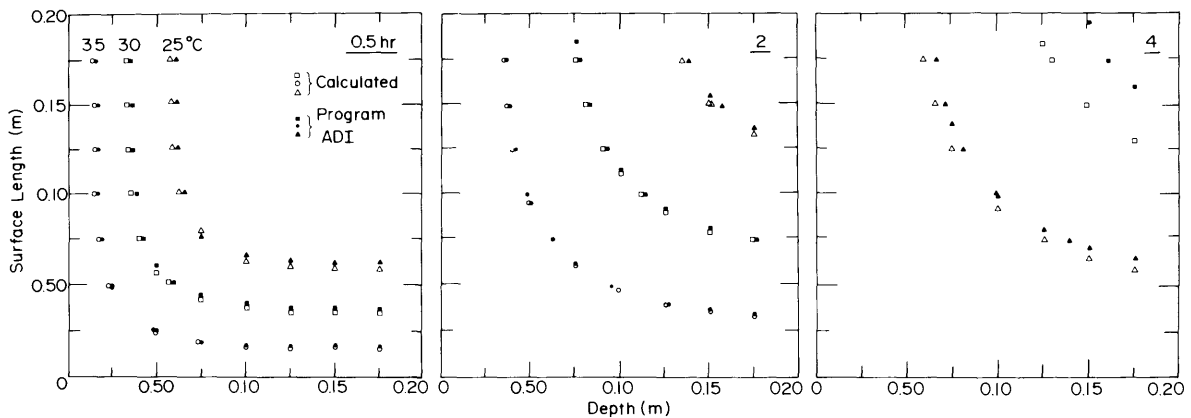


Figure 13. Locations of 35° , 30° and 25°C isotherms from ADI and from the analytical solution for the semi-infinite corner problem.

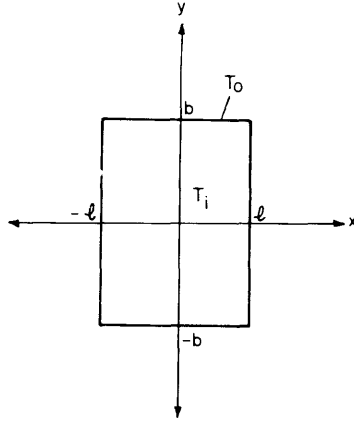


Figure 14. Finite rectangle problem.

Finite rectangle

In order to provide a two-dimensional comparison that doesn't involve the use of semi-infinite boundaries, and also to demonstrate internodal spacing effects, consider the problem of a rectangle of uniform initial temperature T_i that is subject to a step change in the temperatures of all of the edges to T_0 at time zero. The problem is illustrated in Figure 14. The solution of this problem is again obtainable from the product solution of two one-dimensional problems. Carslaw and Jaeger (1959) provide the one dimensional solution:

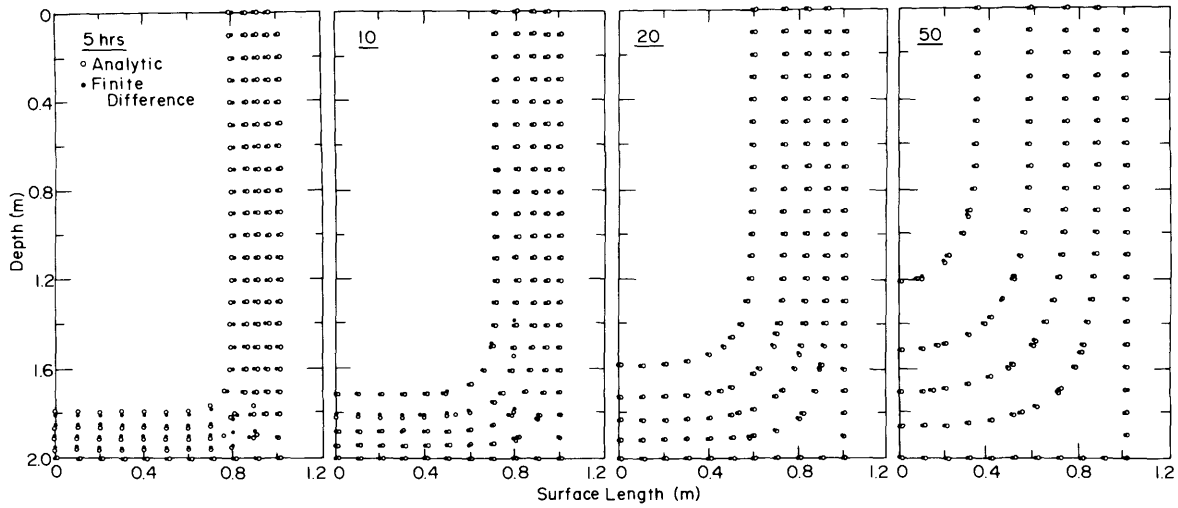
$$\frac{T(x, t) - T_0}{T_i - T_0} = \frac{4}{\pi} \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} \exp \frac{-\alpha (2n+1)^2 \pi^2 t}{4\ell^2} \cos \frac{(2n+1)\pi x}{2\ell} \quad (70)$$

The product solution is then given by

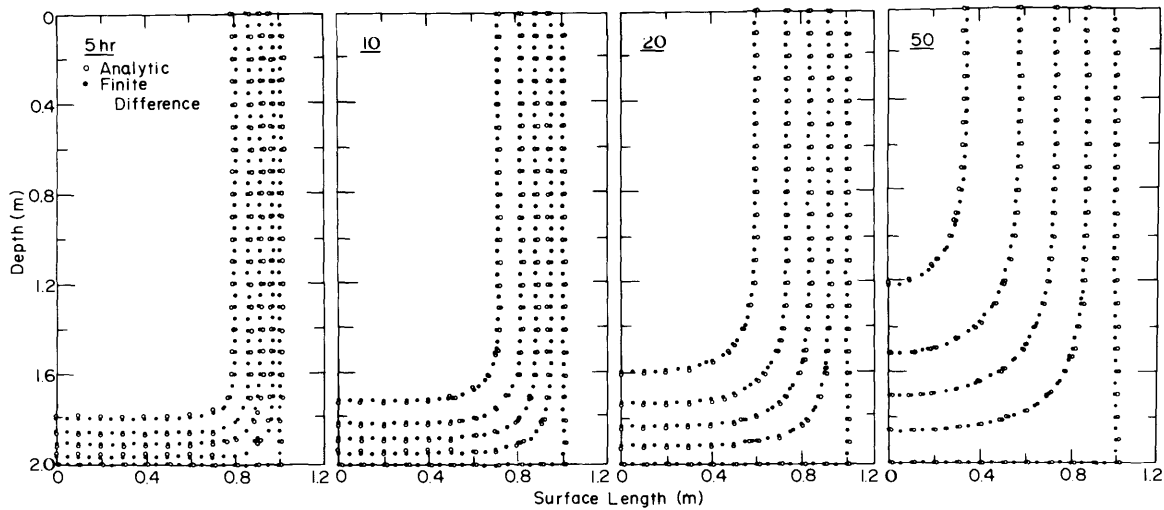
$$T(x, y) = T_0 + \frac{16(T_i - T_0)}{\pi^2} \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} \exp \frac{-\alpha (2n+1)^2 \pi^2 t}{4\ell^2} \cos \frac{(2n+1)\pi x}{2\ell} \cdot \sum_{m=0}^{\infty} \frac{(-1)^m}{2m+1} \exp \frac{-\alpha (2m+1)^2 \pi^2 t}{4b^2} \cos \frac{(2m+1)\pi y}{2b} \quad (71)$$

A short FORTRAN program, INFSUM, was written to compute for this equation the various space and time increments; it is listed with the output in Appendix A. For the comparison, the rectangle is indicated by $\ell = 1$ m, $b = 2$ m, $T_0 = 40^\circ\text{C}$, $T_i = 20^\circ\text{C}$, and the thermal diffusivity α is set equal to 0.00251 m²/hr.

ADI was run several times to demonstrate the effect of internodal spacing on the accuracy of the solution. Because of the symmetry of the problem, we have to model only one quarter of the problem, assigning a zero flux boundary condition to edges of the grid that fall on lines of symmetry. The region modeled is that portion of the rectangle which lies in the fourth quadrant of the Cartesian graph. The first run was made with an internodal spacing of 0.1 m and a time step of 5 hr. The resulting 24° , 28° , 32° , 36° and 40°C isotherms are plotted for several time steps (Fig. 15a). An inspection of the data printed in ADITMP reveals that the maximum discrepancy between the two solutions is 1.5°C , and it occurs in the first time step (5 hr) in the lower right-hand corner. This location at early times represents the steepest temperature gradient in the problem. By the second time step (10 hr), the discrepancy reduces to a maximum of 0.6°C . The maximum difference between the solutions continues to decrease as the temperature gradient decreases



a. Internodal spacing 0.1 m; time step 5 hr.



b. Internodal spacing 0.05 m; time step 2.5 hr.

Figure 15. Locations of 24° , 28° , 32° , 36° and 40° C isotherms (temperature increasing from left to right) from ADI (finite difference) and analytical solution.

and the location of the maximum temperature difference changes to the locations of the steepest gradients. The input and output for this run of ADI are listed in Appendix A.

The same problem was run with an internodal spacing of 0.05 m and a time step of 2.5 hours. The resulting isotherm plots are shown in Figure 15b. A closer agreement was found between the results of ADI and the analytical solution. A node-by-node comparison shows the maximum difference to be 1° C for the 5-hr distribution, decreasing to 0.4° in the 10 hr distribution. In general, the accuracy of the solution is improved as the internodal spacing decreases; the user must determine the accuracy demanded.

One-dimensional semi-infinite problem

It is also of interest to compare the results of ADI to the one-dimensional problem of a medium initially at a uniform temperature; the surface temperature then undergoes a step change to a different temperature, and the resulting temperature distribution is examined over time. This



Figure 16. One-dimensional semi-infinite problem.

problem corresponds to the theoretical setup of the experimental problem examined in the *Comparison of ADI with Experimental Results* section. The problem is illustrated in Figure 16.

The solution to this analytical problem is found in many texts on heat transfer; the reader may refer to Holman (1972). The solution is given as

$$\frac{T(x, t) - T_0}{T_1 - T_0} = \text{erf} \frac{x}{2\sqrt{\alpha t}} \quad (72)$$

where x = the distance below the surface

t = time

α = the thermal diffusivity, 0.00251 m²/hr in this example.

The grid used for this was 3 by 40 nodes, with an internodal spacing of 0.025 m and a time step of 0.25 hr.

For two depths, 0.05 and 0.20 m, the results of this analytical solution are compared to the results of ADI over 9 hr. The results are given in Table 1.

Table 1. Comparison of ADI results to analytical results for the one-dimensional semi-infinite problem.

x (m)	t (hr)	$T(x, t)$	ADI
0.05	0.5	26.42	26.66
	1.0	29.12	29.16
	2.0	31.42	31.41
	3.0	32.51	32.51
	4.0	33.18	33.18
	5.0	33.65	33.65
	6.0	34.00	34.00
	7.0	34.27	34.27
	8.0	34.50	34.49
	9.0	34.68	34.68
0.20	0.5	21.11	21.12
	1.0	21.19	21.21
	2.0	21.88	21.90
	3.0	22.83	22.85
	4.0	23.75	23.76
	5.0	24.56	24.56
	6.0	25.26	25.27
	7.0	28.88	25.88
	8.0	26.42	26.42
	9.0	26.89	26.89

The results compare to within 0.02°C for times after 2 hr. Earlier times have temperatures that differ by as much as 0.24°C near the top of the grid. Probably, this is because the steepest temperature gradients occur in this problem soon after the step change in temperature occurs at the surface. Recall that steeper temperature gradients require a smaller internodal distance; if we want greater accuracy, the model could be again run using an internodal spacing of less than 0.025 m. The smaller the internodal distance, the more accurate the solution.

Comparison of ADI with experimental results

As an example of the use of ADI, let us compare it to some experimental results. Data are available from a full-scale experiment that was conducted in a 4- by 5-m container of soil that was 1.2 m deep. Of the 4-m width, only the center 1.76 m was included in the experiment; the rest was outside an insulated boundary and was put there for support. The soil used was Lebanon sand of 19.7% moisture content. The experiment was designed to start at an initial uniform temperature of 70°F (21.11°C) in the sand in the box, then the surface temperature was to be raised 100°F (37.78°C), and the change in the temperature distribution over time was to be monitored. An analytical solution is available for this case, and ADI was also run for this case. The results compare very well (see the *One-Dimensional Semi-infinite Problem* section, analytical comparisons).

In the comparison of the experimental data to calculated results, temperatures were used that represent a 20-cm wide band taken vertically through the center of the box. Three strings of thermocouples were placed in this band, as shown in Figure 17.

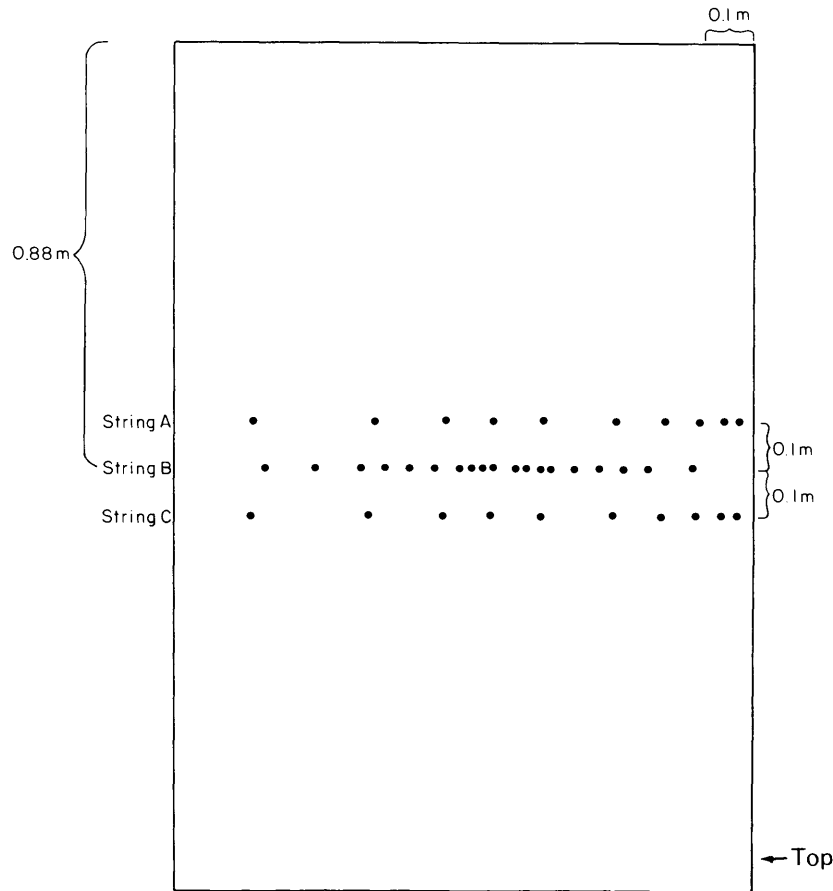


Figure 17. Thermocouple locations in the experimental setup (top of soil container on right).

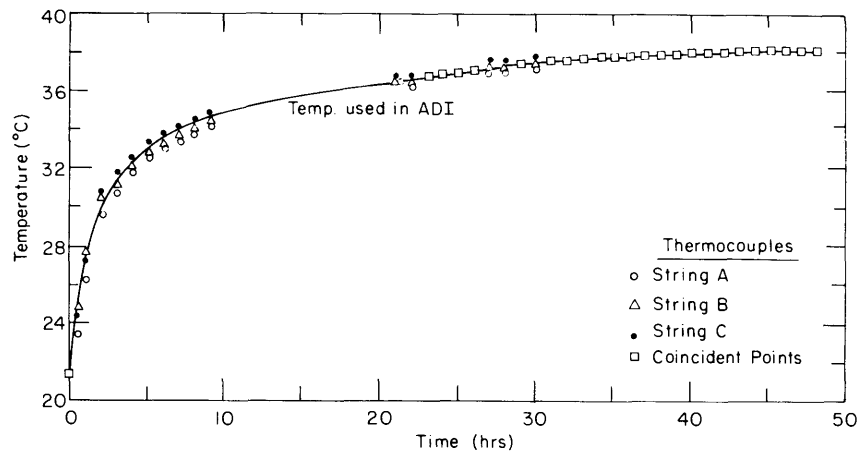


Figure 18. Surface temperatures from experiment used in ADI runs.

In the actual experiment, the initial temperatures ranged from 20.6° to 22.4°C. The initial temperature distribution for ADI was taken as an average of the three thermocouple string temperatures for each depth. It required approximately 27 hr to raise the surface temperature to 37.78°C, after which the surface temperature remained in the range from 37.2° to 38.4°C. Because the data logger malfunctioned, 12 hr of data are missing in a period beginning 9 hr after the start of the experiment; however, the rest of the equipment continued to operate normally. In comparisons made with these data it is assumed that the surface temperature increased linearly over time for this period. See Figure 18 for a graph of the surface temperature vs time for the data and that used in the comparison run with ADI.

Soil tests were conducted at CRREL to determine the density and thermal conductivity of the sand. The density was determined to be 1996.25 kg/m³; this density was assumed constant in the ADI comparison. The conductivity was measured at two temperatures above freezing. At 4.44°C the conductivity was determined to be 1.673 W/m K, and at 26.67°C it was 1.803 W/m K. Each node in the ADI model was assigned a conductivity according to its temperature (in degrees C) for each time step from the following equation:

$$k_{i,j} = 0.00585T + 1.647 . \quad (73)$$

This equation was determined from the two conductivity tests.

The value of the specific heat was taken from measurements done on Lowell sand by Kersten (1949). Again, each node in the ADI model was assigned a temperature-dependent specific heat. The equation fit to Kersten's data is

$$C_{P_{i,j}} = 0.0039T + 0.34927 . \quad (74)$$

The 2 by 48 grid used in ADI had a distance between nodes of 0.025 m. The temperature at the surface was specified, and was taken from the data previously discussed. The time step was 0.25 hr; the surface temperature for each time step was interpolated linearly from the data. The sides and bottom of the grid were assigned the zero heat flux boundary condition. The problem was modeled for 48 hr.

The graphs in Figure 19 compare the results for several times (9 hr, 24 hr and 48 hr) during the run; they show a reasonable agreement but the model tended to underpredict the temperature change.

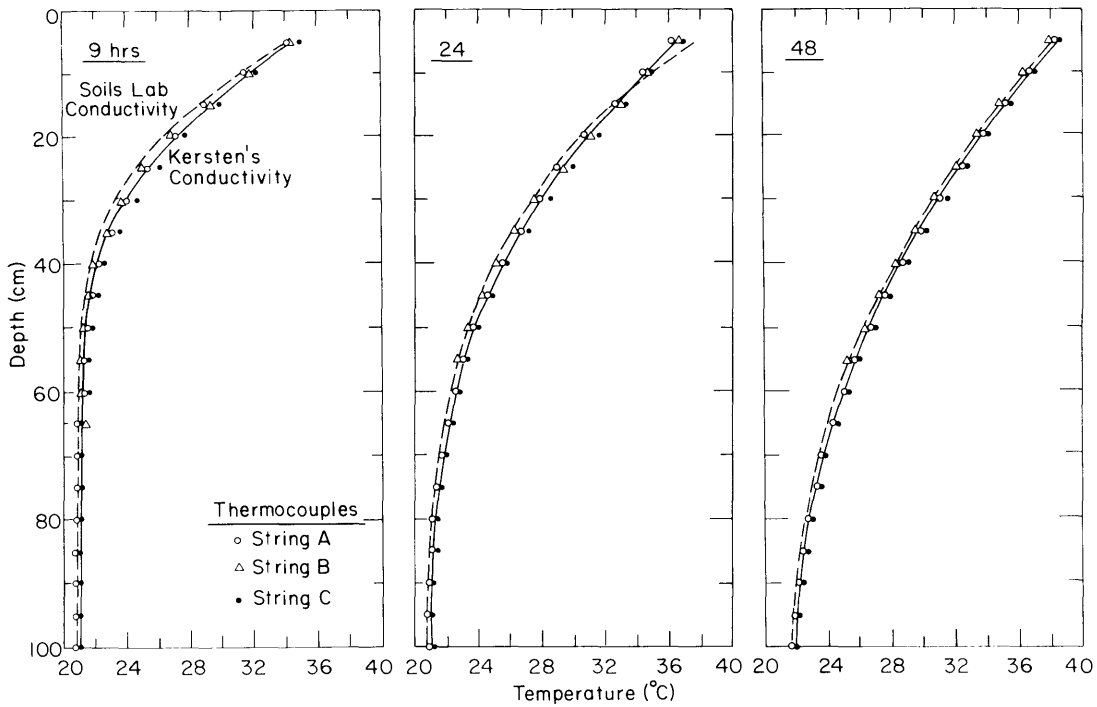


Figure 19. Results of two ADI runs using different conductivities plotted against experimental results.

The model was then run again, using conductivity measurements on Lowell sand by Kersten (1949). The temperature-dependent equation for conductivity used for this run was

$$k_{i,j} = 0.00585T + 1.975 . \quad (75)$$

These results may again be found in Figure 19. This time excellent agreement was found between the actual data and the ADI calculations.

Each of the two runs of ADI required 2 minutes and 13 seconds of computer time on the PRIME computer at CRREL.

VERIFICATION OF ADIPC

Comparison of ADIPC with analytical results—the Neumann solution

ADIPC was first compared to the well-known one-dimensional analytical Neumann solution. A semi-infinite region is initially at a uniform temperature T_0 which is above the fusion temperature T_f . Suddenly, the surface temperature is changed to T_s , a temperature below the fusion temperature. The subsequent movement of the phase change front may be calculated.

Allow ρ , C_p , k and H_L to represent the density, specific heat, thermal conductivity and latent heat of fusion, respectively. Thermal diffusivity α is defined as $k/\rho C_p$. Temperature T is a function of position and time. Subscripts 1 and 2 refer to the property or variable in the frozen and liquid phase, respectively. The boundary conditions follow:

$$T_1(0, t) = T_s \quad (76)$$

$$T_2(x, 0) = T_0 \quad (77)$$

$$T_1(X, t) = T_2(X, t) = T_f \quad (78)$$

$$k_1 \frac{\partial T_1}{\partial X} - k_2 \frac{\partial T_2}{\partial X} = \rho H_L \frac{\partial X}{\partial t} \quad (79)$$

The temperatures in the liquid and solid regions must satisfy the following equations

$$\frac{\partial^2 T_1}{\partial x^2} - \frac{1}{k_1} \frac{\partial T_1}{\partial x} = 0 \quad (80)$$

$$\frac{\partial^2 T_2}{\partial x^2} - \frac{1}{k_2} \frac{\partial T_2}{\partial x} = 0 \quad (81)$$

The solution has been given many times in the literature (e.g. Carslaw and Jaeger 1959). The solution for the location of the phase change front follows:

$$X = 2\lambda \sqrt{\alpha_1 t} \quad (82)$$

where λ must be determined from the relation

$$\frac{e^{-\lambda^2}}{\operatorname{erf} \lambda} \frac{k_2 \sqrt{\alpha_1} (T_0 - T_f) e^{-\alpha_1/\alpha_2 \lambda^2}}{k_1 \sqrt{\alpha_2} (T_f - T_s) \operatorname{erfc} \lambda \sqrt{\alpha_1/\alpha_2}} = \frac{\lambda H_L \sqrt{\pi}}{C_{P1} (T_f - T_s)} \quad (83)$$

For comparison the following values were used:

$$k_1 = 2.21 \frac{\text{W}}{\text{m K}} \quad k_2 = 0.580 \frac{\text{W}}{\text{m K}}$$

$$\rho_1 = 917 \frac{\text{kg}}{\text{m}^3} \quad \rho_2 = 998.2 \frac{\text{kg}}{\text{m}^3}$$

$$C_{P1} = 0.5815 \frac{\text{W hr}}{\text{kg K}} \quad C_{P2} = 1.16 \frac{\text{W hr}}{\text{kg K}}$$

$$H_L = 93.00 \frac{\text{W hr}}{\text{kg}}$$

$$T_f = 0^\circ\text{C}$$

$$T_s = -4.67^\circ\text{C}$$

$$T_0 = 4.67^\circ\text{C}$$

The problem was modeled for 3 hr using an internodal spacing of 0.5 cm and a time step of 0.0025 hr. The depth of the 0°C isotherm is plotted with the analytical solution in Figure 20. As mentioned earlier, the location of the phase change front is found by interpolating between the nodal temperatures to find the 0°C isotherm. The front progresses in a step-like pattern. This occurs when the location of phase change moves from one node to the next and is inherent in the apparent heat capacity method in a discretized space. Nevertheless, the results of ADIPC show good agreement with the analytical solution. A more accurate computed solution could be obtained by using a smaller internodal spacing and smaller time step. A copy of ADIPC and output for this solution is included in Appendix B.

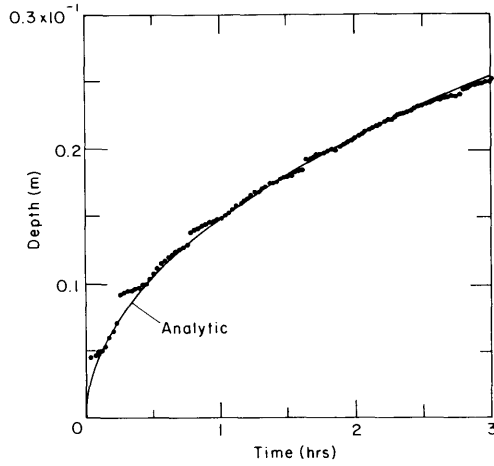


Figure 20. Comparison of ADIPC with Neumann solution.

Comparison of ADIPC with analytical results—two-dimensional phase change verification

Next compare the results of ADIPC with an analytical solution involving phase change around a pipe. This problem is two-dimensional in Cartesian coordinates and thus is a two-dimensional verification of ADIPC. But it is a one-dimensional problem in cylindrical coordinates, facilitating an analytical solution. An exact analytical solution is available for the case of freezing in a region, initially at a uniform temperature, that is suddenly subject to the effects of a continuous line source that extracts heat at a rate of Q per unit time. The exact solution of this heat conduction problem is given in Carslaw and Jaeger (1959). The results are as follows.

The location of the freezing front at any time is given by

$$R = 2\lambda \sqrt{\alpha_1 t} \quad (84)$$

where R is the radius of the front ($r = 0$ is the location of the line source), α_1 is the thermal diffusivity of the frozen zone, t is time and λ is obtained from the following relation:

$$\frac{Q}{4\pi} \exp(-\lambda^2) + \frac{k_2 (T_i - T_f)}{E_i(-\lambda \alpha_1 / \alpha_2)} \exp(-\lambda^2 \alpha_1 / \alpha_2) = \lambda^2 x_1 L\rho \quad (85)$$

where k = conductivity
 α = diffusivity
 T_i = initial temperature
 T_f = fusion temperature
 $L\rho$ = latent heat per unit volume.

E_i is the exponential integral function

$$E_i(x) = \int_{-\infty}^x \frac{e^v dV}{V},$$

and the subscripts 1 and 2 represent the frozen and unfrozen zones, respectively.

The temperatures in the frozen and unfrozen zones are given by

$$T_1 = T_f + \frac{Q}{4\pi k_1} \left[E_i\left(-\frac{r^2}{4\alpha_1 t}\right) - E_i(-\lambda^2) \right] \quad 0 < r < R \quad (86)$$

$$T_2 = T_i - \frac{(T_i - T_f)}{E_i (-\lambda^2 \alpha_1/\alpha_2)} E_i \left(\frac{-r^2}{4\alpha_2 t} \right) \quad r > R . \quad (87)$$

In order to get a solution for phase change around a pipe from this, the pipe is assigned a constant radius r_p ; the temperature at r_p varies with time according to eq 86. A shifted time is used so that at time $t = 0$ in the computer run, the location of the phase change front is at $r = r_p$. For this time, the initial temperature distribution is figured from eq 87.

For the comparison, the following values were used:

$$k_1 = 0.0072 \frac{\text{cal}}{\text{cm s } ^\circ\text{C}} \quad k_2 = 0.0042 \frac{\text{cal}}{\text{cm s } ^\circ\text{C}}$$

$$\alpha_1 = 0.014165 \frac{\text{cm}^2}{\text{s}} \quad \alpha_2 = 0.005556 \frac{\text{cm}^2}{\text{s}}$$

$$L\rho = 33.012 \frac{\text{cal}}{\text{cm}^2 \text{ } ^\circ\text{C}}$$

$$T_i = 4^\circ\text{C}$$

$$T_f = 0^\circ\text{C} .$$

Densities for all regions were assumed equal in the solution.

Equation 84 was solved by substituting a polynomial approximation for the exponential integrals (Abramowitz and Stegun 1970) and then solving the equation for λ by an iterative scheme on the computer. For this case it was found that $\lambda = 0.08246$.

ADIPC used an internodal distance of 1 cm and a time step of 60 s on a 30 by 30 grid. Because of the symmetry of the problem, with the Cartesian origin at the center of the pipe, only one-quarter of the situation was modeled. The top and right-hand edges of the grid were assigned zero flux boundaries, and the left-hand and bottom edges of the grid were assigned the semi-infinite condition.

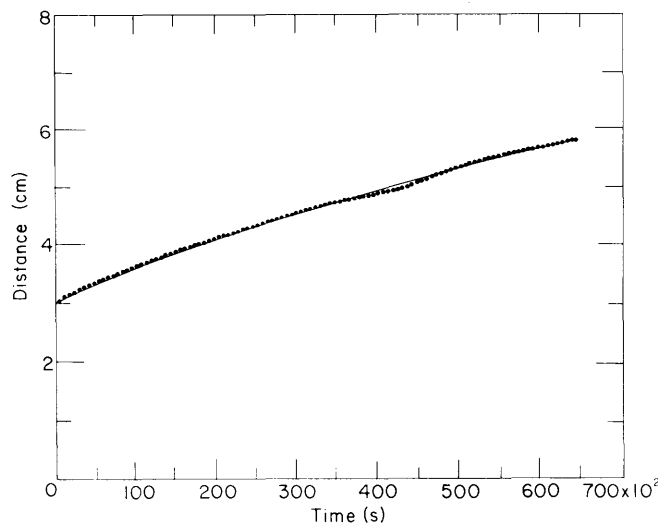


Figure 21. Comparison of ADIPC with radial analytic freezing solution.

The location of the freezing front over time was found in ADIPC by interpolating to find the zero degree isotherm; this is plotted against the analytical solution in Figure 21. Excellent agreement is found.

USER INSTRUCTION FOR ADI

The first step in using ADI to solve a problem is to define the boundaries of the problem and to identify the different materials in the problem. It is important to know the dimensions of objects to be modeled as accurately as possible. Next, identify the boundary conditions in the problem. Now draw the grid, determining a nodal spacing that will accurately represent the materials in the problem. Indicate the boundary conditions on the grid, then set these conditions in the computer program by assigning the appropriate values of RAY (I, J, 2) to each node I, J in the grid; the values are listed in the comment statements at the beginning of subroutine ADDATA.

Material properties such as thermal conductivity, density, etc., must be known in consistent units. There are no dimensional constants inherent in the program, so any system may be used. For the problems presented in this report, the following units were used: thermal conductivity (W/m K), density (kg/m^3), specific heat (W hr/kg K), temperature ($^{\circ}\text{C}$), time (hr) and distance (m). The units of the convection coefficient would be $\text{W/m}^2\text{K}$. Values for the conductivity, specific heat, density and convection coefficient must be specified by the user as indicated in the main program (near the start of the 2001 and 2002 loops). These values may be programmed to change with temperature or time and could conceivably be different for each node in the grid. At the time of this report's publication, the conductivity of a semi-infinite node in the program is assumed equal to that of the adjacent regular interior node.

Now edit subroutine ADDATA following the directions in the comment statements in the subroutine to initialize the variables and arrays. Note that the variables and arrays are defined at the beginning of ADDATA.

When editing subroutine ADDATA, the user will encounter a variable named ITRT, which is defined as the number of time steps before the results are printed. Setting this value as five, for example, will set a counter in the main program so that at every fifth complete time step the main program will print the temperature distribution (RAY [I, J, 1]) into file ADITMP. Similarly, variable ITPC will set a counter to call subroutine ISOTHM to locate the user-specified isotherms in the current temperature distribution. The coordinates of these isotherms are printed into file POINTS.

When ADI is run, it will interactively ask whether or not the user wishes to run subroutine ADDATA. The first time the program is run, ADDATA must be run. The variables will be put into a formatted data file, ADIDAT. If, after the program is run, a change is made in the main program but not to ADDATA, the subroutine need not be called again; the main program will read the data from ADIDAT.

Once the program has been run, the user should examine output file ADIOUT closely to be sure that the initial conditions and boundary conditions are those intended, i.e., that no mistake was made when editing ADIDAT or changing values in the main program.

USER INSTRUCTION FOR ADIPC

The use of ADIPC is the same as ADI with the exception of the specification of several phase change variables. When editing ADDATA, the user must specify the value of the latent heat of fusion, the conductivity and density of the phase change region, and the temperature range over which phase change will occur. If the units indicated in the instructions for ADI are used, the

latent heat will have the units W hr/kg, as does the specific heat. The value of the apparent specific heat (discussed in the *Phase Change* section), known as CPPC in the program, should be calculated by the user as follows:

$$CPPC = \frac{1}{2} (C_{PB} + C_{PA}) + \frac{H_L}{\Delta T}$$

where C_{PB} = specific heat of the frozen material

C_{PA} = specific heat of the unfrozen material

H_L = latent heat of fusion for the material

ΔT = temperature range around the fusion temperature where phase change occurs from

$$T_f - \frac{\Delta T}{2} \text{ to } T_f + \frac{\Delta T}{2} .$$

ΔT in the verifications was 1.0°C.

As in ADI, the values for density, specific heat and conductivity must be specified as indicated in the comments near the start of the main program, ADIPC.

CONCLUSIONS

Two two-dimensional finite difference computer programs have been developed to model time-varying heat conduction. Results of test runs of the programs show excellent agreement with analytical and experimental results. The programs are easily set up to model new problems, and have the capability to solve a wide variety of heat conduction problems.

LITERATURE CITED

- Abramowitz, M. and I.A. Stegun** (1970) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards.
- Bonacina, C. and Comini, G.** (1973) On the solution of the nonlinear heat conduction equations by numerical methods. *International Journal of Heat and Mass Transfer*, **16**: 581-589.
- Carnahan, B., H.A. Luther and J.O. Wilkes** (1969) *Applied Numerical Methods*. New York: John Wiley and Sons, Inc.
- Carslaw, H.S. and J.C. Jaeger** (1959) *Conduction of Heat in Solids*. Oxford: The Clarendon Press.
- Croft, D.R. and D.G. Lilley** (1977) *Heat Transfer Calculations Using Finite Difference Equations*. London: Applied Science Publishers, Ltd.
- Gerald, C.F.** (1980) *Applied Numerical Analysis*. Reading, Massachusetts: Addison-Wesley.
- Holman, J.P.** (1972) *Heat Transfer*, 3rd ed. New York: McGraw-Hill.
- Peaceman, D.W. and H.H. Rachford** (1955) The numerical solution of parabolic and elliptic differential equations. *Journal of the Society of Industrial Applied Mathematicians*, **3**: 28-41.
- Phetteplace, G.L., W. Wiley and M.A. Novick** (1981) Losses from the Fort Wainwright heat distribution system. CRREL Special Report 81-14.

APPENDIX A: PROGRAM INFSUM AND SAMPLE INPUT AND OUTPUT FOR PROGRAM ADI

INFSUM, a short program to do infinite sum calculation for rectangle problem

```

C  INFSUM
C  THIS CALCULATES THE INFINITE SUM PRODUCT FOR THE RECTANGLE.
    IMPLICIT DOUBLE PRECISION(A-H,O-Z)
    IMPLICIT INTEGER*4(I-N)
    DIMENSION TEMP(50,50)
    CALL CTRL(2,'INFOUT',5)
    N=50
    A=.00251
    EL=1
    B=2
    T0=40.
    T1=20.0
    PI=3.141592654
    PI2=9.869604404
    DS=.1
    DELT=1
    IX=11
    IY=21
    DO 100 IT=5,M,5
    TIM=IT*DELT
    DO 200 I=1,IY
    DO 210 J=1,IX
    Y=DS*(I-1.)
    X=DS*(J-1.)
    C1=-A*PI2*TIM/(4.*EL*EL)
    C2=PI*X/(2.*EL)
    C3=-A*PI2*TIM/(4.*B*B)
    C4=PI*Y/(2.*B)
C      FIGURE SUM1
    N=1
    SUM1=0.
    DO 220 INN=1,50
    IN=INN-1
    C1A=C1*(2.*IN+1.)*(2.*IN+1.)
    C1AA=DEXP(C1A)
    C2A=C2*(2.*IN+1.)
    FRST=((N/(2.*IN+1.))*C1AA*DCOS(C2A))
    SUM1=SUM1+FRST
    N=-1*N
    IF(C1AA.LE..0001) GO TO 221
220  CONTINUE
221  CONTINUE
C      FIGURE SUM2
    N=1
    SUM2=0.
    DO 230 INN=1,50
    IN=INN-1
    C3A=C3*(2.*IN+1.)*(2.*IN+1.)
    C4A=C4*(2.*IN+1.)
    C3AA=DEXP(C3A)
    SCND=(N/(2.*IN+1.))*C3AA*DCOS(C4A)
    SUM2=SUM2+SCND
    N=-1*N
    IF(C3AA.LE..0001) GO TO 231
230  CONTINUE
231  CONTINUE
    TEMP(I,J)=T0+(T1-T0)*(16./PI2)*SUM1*SUM2
210  CONTINUE
200  CONTINUE
    WRITE(5,211) TIM
211  FORMAT(/,1X,'TEMPERATURES AFTER',F6.2,' HOURS')
    WRITE(5,212) ((TEMP(I,J),J=1,IX),I=1,IY)
212  FORMAT(1X,11F6.2)
    CONTINUE
100  CALL CTRL(4,'INFOUT',5)
    CALL EXIT
    END

```


34.00	DEGREE	ISOTHERM:
0.049		0.100
0.049		0.200
0.049		0.300
0.049		0.400
0.049		0.500
0.049		0.600
0.049		0.700
0.049		0.800
0.049		0.900
0.049		1.000
0.049		1.100
0.049		1.200
0.049		1.300
0.049		1.400
0.049		1.500
0.049		1.600
0.049		1.700
0.049		1.800
0.049		1.900
0.049		2.000
0.049		2.100
0.049		2.200
0.049		2.300
0.049		2.400
0.049		2.500
0.049		2.600
0.049		2.700
0.049		2.800
0.049		2.900
0.049		3.000
0.049		3.100
0.049		3.200
0.049		3.300
0.049		3.400
0.049		3.500
0.049		3.600
0.049		3.700
0.049		3.800
0.049		3.900
0.049		4.000
0.049		4.100
0.049		4.200
0.049		4.300
0.049		4.400
0.049		4.500
0.049		4.600
0.049		4.700
0.049		4.800
0.049		4.900
0.049		5.000
0.049		5.100
0.049		5.200
0.049		5.300
0.049		5.400
0.049		5.500
0.049		5.600
0.049		5.700
0.049		5.800
0.049		5.900
0.049		6.000
0.049		6.100
0.049		6.200
0.049		6.300
0.049		6.400
0.049		6.500
0.049		6.600
0.049		6.700
0.049		6.800
0.049		6.900
0.049		7.000
0.049		7.100
0.049		7.200
0.049		7.300
0.049		7.400
0.049		7.500
0.049		7.600
0.049		7.700
0.049		7.800
0.049		7.900
0.049		8.000
0.049		8.100
0.049		8.200
0.049		8.300
0.049		8.400
0.049		8.500
0.049		8.600
0.049		8.700
0.049		8.800
0.049		8.900
0.049		9.000
0.049		9.100
0.049		9.200
0.049		9.300
0.049		9.400
0.049		9.500
0.049		9.600
0.049		9.700
0.049		9.800
0.049		9.900
0.049		10.000
32.00	DEGREE	ISOTHERM:
0.098		0.100
0.098		0.200
0.098		0.300
0.098		0.400
0.098		0.500
0.098		0.600
0.098		0.700
0.098		0.800
0.098		0.900
0.098		1.000
0.098		1.100
0.098		1.200
0.098		1.300
0.098		1.400
0.098		1.500
0.098		1.600
0.098		1.700
0.098		1.800
0.098		1.900
0.098		2.000
0.098		2.100
0.098		2.200
0.098		2.300
0.098		2.400
0.098		2.500
0.098		2.600
0.098		2.700
0.098		2.800
0.098		2.900
0.098		3.000
0.098		3.100
0.098		3.200
0.098		3.300
0.098		3.400
0.098		3.500
0.098		3.600
0.098		3.700
0.098		3.800
0.098		3.900
0.098		4.000
0.098		4.100
0.098		4.200
0.098		4.300
0.098		4.400
0.098		4.500
0.098		4.600
0.098		4.700
0.098		4.800
0.098		4.900
0.098		5.000
0.098		5.100
0.098		5.200
0.098		5.300
0.098		5.400
0.098		5.500
0.098		5.600
0.098		5.700
0.098		5.800
0.098		5.900
0.098		6.000
0.098		6.100
0.098		6.200
0.098		6.300
0.098		6.400
0.098		6.500
0.098		6.600
0.098		6.700
0.098		6.800
0.098		6.900
0.098		7.000
0.098		7.100
0.098		7.200
0.098		7.300
0.098		7.400
0.098		7.500
0.098		7.600
0.098		7.700
0.098		7.800
0.098		7.900
0.098		8.000
0.098		8.100
0.098		8.200
0.098		8.300
0.098		8.400
0.098		8.500
0.098		8.600
0.098		8.700
0.098		8.800
0.098		8.900
0.098		9.000
0.098		9.100
0.098		9.200
0.098		9.300
0.098		9.400
0.098		9.500
0.098		9.600
0.098		9.700
0.098		9.800
0.098		9.900
0.098		10.000

APPENDIX B: PROGRAM ADIPC AND SAMPLE INPUT AND OUTPUT

Program ADIPC and its subroutines

```
1)
2) C *****
3) C *****
4) C ADIPC
5) C
6) C THIS FORTRAN PROGRAM SOLVES FOR TWO DIMENSIONAL TEMPERATURE
7) C DISTRIBUTION RESULTING FROM CONDUCTION HEAT TRANSFER. BOUNDARY CONDITIONS
8) C MAY INCLUDE SPECIFIED TEMPERATURES, CONVECTIVE SURFACES, AND
9) C SEMI-INFINITE BOUNDARIES. ADIPC USES AN IMPLICIT ALTERNATING DIRECTION
10) C FINITE DIFFERENCE NUMERICAL TECHNIQUE ON A GRID WITH SQUARE ELEMENTS.
11) C THE MATERIAL PROPERTIES (TK,CP,RO,H) MAY BE DIFFERENT FOR EACH
12) C NODE OF THE GRID, AND MAY CHANGE WITH TIME.
13) C ADIPC USES THE APPARENT HEAT CAPACITY METHOD FOR PHASE CHANGE.
14) C THIS PROGRAM WAS WRITTEN BY MARY REMLEY ALBERT AT CRREL, 1981.
15) C
16) C DATA FOR ADIPC IS GATHERED BY ADDATA, WHICH PUTS IT INTO FILE ADPDAT.
17) C SEE ADDATA FOR AN EXPLANATION OF VARIABLES, AND TO SET UP THE PROGRAM
18) C FOR YOUR PROBLEM.
19) C ADJUSTMENTS FOR CHANGING MATERIAL PROPERTIES WITH TIME (DURING THE RUN)
20) C SHOULD BY TYPED INTO THE MAIN PROGRAM AS INDICATED NEAR LINE 170.
21) C
22) C *****
23) C *****
24) C
25) C
26) C DO NOT CHANGE THE NEXT 130 LINES.
27) C IMPLICIT INTEGER*2(A,B,I-L,N,Q,W,X,Y,Z)
28) C IMPLICIT DOUBLE PRECISION(C-H,M,O,P,R-V)
29) C M1 IS THE COMMON LOCATION FOR FOR MAIN AND ADDATA VARIABLES
30) C M12 IS THE COMMON LOCATION FOR FOR MAIN, ADDATA, AND ISOTHM VARIABLES
31) C COMMON/M12I/ X,Y,NISO,Q
32) C COMMON/M12R/ RAY(80,80,3),TISO(9),DS,DELT
33) C COMMON/M11/ IMAX,A,ITRT,KRNR(4),ITPC
34) C COMMON/M1R/ DI,TDEL,H(2),FLXT(80),
35) C & FLXB(80),FLXL(80),FLXR(80),TMPT(80),
36) C & TMPB(80),TMPL(80),TMPR(80),ROPC(2),CPPC(2),HL(2),TPC(2)
37) C COMMON/MMM/ TEMP(80,80),EA(200),
38) C & EB(200),C(200),D(200),CP(80,80),TK(80,80),
39) C & RO(80,80),OLDT(80,80),CPO(80,80),ROO(80,80),
40) C & IFLAG(10),ISTAT(80,80)
41) C OPEN FILES
42) C CALL CONTRL(3,'ADPDAT',5)
43) C CALL CONTRL(2,'ADPOUT',6)
44) C FUNITS 7 & 8 USED IN ISOTHM. FUNIT 10 NOT USABLE
45) C CALL CONTRL(2,'ADPNDT',9)
46) C CALL CONTRL(2,'ADPTMP',11)
47) C CALL CONTRL(2,'POINT1',13)
48) C WRITE(1,1)
49) 1 FORMAT(1X,'IF YOU CHANGED ADDATA SINCE YOU LAST RAN THIS,',
50) C 'TYPE "1"',/,1X,'AND ADDATA WILL BE EXECUTED.',/,1X,
51) C 'OTHERWISE, TYPE "2"')
52) C READ(1,*) IT
53) C IF(IT.EQ.2) GO TO 4
54) C CALL ADDATA
55) C GO TO 12
56) 4 CONTINUE
57) C READ DATA INPUT FROM FILE ADPDAT
58) 10 READ(5,10) DS,DELT,DI,TDEL
59) 10 FORMAT(1X,4F10.5)
60) 20 READ(5,20) A,X,Y,NISO,ITRT,IMAX,ITPC
61) 20 FORMAT(1X,7I5)
62) 16 READ(5,16) ((ROPC(K),CPPC(K),HL(K),TPC(K)),K=1,A)
63) 16 FORMAT(1X,4F10.5)
64) 15 READ(5,15)(TISO(B),B=1,NISO)
65) 15 FORMAT(1X,F7.2)
66) 30 READ(5,30) (H(L),L=1,A)
67) 30 FORMAT(1X,F10.5)
68) 34 READ(5,35) ((RAY(I,J,3),J=1,X),I=1,Y)
69) 34 READ(5,35) ((RAY(I,J,2),J=1,X),I=1,Y)
70) 34 READ(5,35) ((RAY(I,J,1),J=1,X),I=1,Y)
71) 35 FORMAT(1X,17F7.2)
72) 34 READ(5,34) (KRNR(J),J=1,4)
73) 34 FORMAT(1X,4I2)
74) 34 READ(5,35) (FLXT(J),J=1,X)
75) 34 READ(5,35) (FLXB(J),J=1,X)
76) 34 READ(5,35) (FLXL(I),I=1,Y)
77) 34 READ(5,35) (FLXR(I),I=1,Y)
78) 34 READ(5,35) (TMPT(J),J=1,X)
79) 34 READ(5,35) (TMPB(J),J=1,X)
80) 34 READ(5,35) (TMPL(I),I=1,Y)
```

```

81) READ(5,35) (TMPR(I),I=1,Y)
82) CONTINUE
83) C WRITE INITIAL DATA INTO OUTPUT FILE ADPOUT
84) WRITE(6,21)
85) 21 FORMAT(/,1X,'DATA FOR THIS RUN OF ADIPC: ',/,/)
86) WRITE(6,22)
87) 22 FORMAT(/,1X,' DS DELT DI TDEL')
88) WRITE(6,10) (DS,DELT,DI,TDEL)
89) WRITE(6,23)
90) 23 FORMAT(/,5X,'A',4X,'X',4X,'Y',1X,'NISO ITRT IMAX ITPC ')
91) WRITE(6,20) (A,X,Y,NISO,ITRT,IMAX,ITPC)
92) WRITE(6,29)
93) 29 FORMAT(/,1X,' ROPC(K) CPPC(K) HL(K) TPC(K)')
94) WRITE(6,16) ((ROPC(K),CPPC(K),HL(K),TPC(K)),K=1,A)
95) WRITE(6,24)
96) 24 FORMAT(/,1X,'TISO(B),B=1,NISO:')
97) WRITE(6,15) (TISO(B),B=1,NISO)
98) WRITE(6,25)
99) 25 FORMAT(/,3X,'H(K)')
100) WRITE(6,30) (H(L),L=1,A)
101) WRITE(6,33) (KRNR(J),J=1,4)
102) 33 FORMAT(/,1X,'KRNR(1)=',I2,' KRNR(2)=',I2,' KRNR(3)=',I2,
103) & ' KRNR(4)=',I2)
104) WRITE(6,26)
105) 26 FORMAT(/,3X,'FLXT(J)',3X,'FLXB(J)',3X,'TMPT(J)',3X,
106) & 'TMPB(J)')
107) WRITE(6,27) ((FLXT(J),FLXB(J),TMPT(J),TMPB(J)),J=1,X)
108) 27 FORMAT(1X,4F10.2)
109) WRITE(6,28)
110) 28 FORMAT(/,3X,'FLXL(I)',3X,'FLXR(I)',3X,'TMPL(I)',3X,
111) & 'TMPR(I)')
112) WRITE(6,27) ((FLXL(I),FLXR(I),TMPL(I),TMPR(I)),I=1,Y)
113) RRX=X/17
114) JJX=RRX
115) JJR=X-(17*JJX)
116) DO 63 K=1,3
117) GO TO (64,65,66),K
118) 64 WRITE(6,31)
119) GO TO 67
120) 65 WRITE(6,45)
121) GO TO 67
122) 66 WRITE(6,44)
123) 67 DO 68 JJ=1,JJX
124) IF(JJX.EQ.0) GO TO 68
125) J2=17*JJ
126) J1=J2-16
127) WRITE(6,500) ((RAY(I,J,K),J=J1,J2),I=1,Y)
128) IF(JJ.LT.JJX) WRITE(6,503)
129) 68 CONTINUE
130) J2=X
131) J1=X-JJR+1
132) IF(JJR.EQ.0) GO TO 69
133) IF(JJX.NE.0) WRITE(6,503)
134) DO 69 I=1,Y
135) WRITE(6,500) (RAY(I,J,K),J=J1,J2)
136) 69 CONTINUE
137) 63 CONTINUE
138) 44 FORMAT(/,1X,'RAY(I,J,3) NODAL MATERIAL TYPE')
139) 45 FORMAT(/,1X,'RAY(I,J,2) NODAL LOCATION TYPE')
140) 31 FORMAT(/,1X,'RAY(I,J,1) TEMPERATURES AT THE START:')
141) WRITE(1,11) ITRT,ITPC
142) WRITE(6,11) ITRT,ITPC
143) 11 FORMAT(/,/,1X,'TEMPERATURES WILL BE PRINTED EVERY',I4,
144) & ' TIME STEPS.',/,1X,'ISOTHERMS WILL BE LOCATED EVERY',I4,
145) & ' TIME STEPS.')
146) DO 37 K=1,9
147) IFLAG(K)=0
148) 37 CONTINUE
149) KONT=0
150) JFLG=1
151) Q=0
152) ITCC=0
153) ITPP=0
154) DI2=2.D0*DI-1.D0
155) KDX=1
156) C
157) C
158) C
159) C XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
160) C XXXXXXXXXXXXX DON'T CHANGE ANYTHING ABOVE THIS LINE. XXXXXXXXXXXXXXXXXXXXXXX
161) C INITIALIZE SPECIFIC HEAT AND CONDUCTIVITY
162) C IF TEMPERATURE DEPENDENT, INSERT EQUATIONS.
163) DO 46 J=1,X
164) DO 47 I=1,Y
165) TEMP(I,J)=RAY(I,J,1)
166) K=RAY(I,J,3)
167) IF(RAY(I,J,1).LT.TPC(K)) GO TO 39
168) C UNFROZEN
169) CP(I,J)=1.16000
170) RO(I,J)=998.2000
171) GO TO 47
172) 39 CONTINUE
173) C FROZEN

```

```

174)      CP(I,J)=.5815D0
175)      RO(I,J)=917.0D0
176)  47  CONTINUE
177)  46  CONTINUE
178) C *****
179) C 2002 CONTINUE
180) C          COMES HERE AFTER EVERY COMPLETE TIME STEP
181) C          NOTE THAT Q=0 FOR 1ST ITERATION
182) C
183) C IF SPECIFIED TEMPERATURES CHANGE WITH TIME, SPECIFY THE
184) C CHANGES HERE. (NUMBERS 57 AND 58 ARE FREE FOR LOOPS IF NEEDED.)
185) C
186) C
187) C ADJUST THERMAL PROPERTIES WITH TEMPERATURE.
188) C IF APPROPRIATE, USE RAY(I,J,3) TO INDICATE MATERIAL TYPE.
189) C (BUT DO NOT CHANGE THE NEXT 9 LINES.)
190)      DO 50 J=1,X
191)      DO 51 I=1,Y
192)      K=RAY(I,J,3)
193)      CPO(I,J)=CP(I,J)
194)      ROO(I,J)=RO(I,J)
195)      CC=TPC(K)+(TDEL/2.D0)
196)      DD=TPC(K)-(TDEL/2.D0)
197)      IF(RAY(I,J,1).LT.DD) GO TO 60
198)      IF(RAY(I,J,1).GT.CC) GO TO 61
199) C          UNDERGOING PHASE CHANGE
200)      ISTAT(I,J)=3
201)      TK(I,J)=2.210D0
202)      GO TO 51
203)  60  CONTINUE
204) C          FROZEN
205)      ISTAT(I,J)=1
206)      TK(I,J)=2.210D0
207)      RO(I,J)=917.0D0
208)      CP(I,J)=.5815D0
209)      GO TO 51
210)  61  CONTINUE
211) C          UNFROZEN
212)      ISTAT(I,J)=2
213)      TK(I,J)=.580D0
214)      RO(I,J)=998.2D0
215)      CP(I,J)=1.1630D0
216)  51  CONTINUE
217)  50  CONTINUE
218) C
219) C
220) C XXXXXXXXXXXXXXXXXXXX DON'T CHANGE ANYTHING BELOW THIS LINE XXXXXXXXXXXXXXXXXXXXXXXX
221) C XXXXXXXXXXXXXXXXXXXX UNLESS YOU KNOW WHAT YOU ARE DOING. XXXXXXXXXXXXXXXXXXXXXXXX
222) C 2001 CONTINUE
223) C          COMES HERE AFTER EACH PASS
224)      IF(JFLG.EQ.1) GO TO 161
225) C          FOR 2ND PASS
226)      DO 1600 J=1,X
227)      KONT=0
228)      DO 41 K=1,Y
229)      EA(K)=0.0D0
230)      EB(K)=0.0D0
231)      C(K)=0.0D0
232)      D(K)=0.0D0
233)      O(K)=0.0D0
234)  41  CONTINUE
235)      DO 1700 I=1,Y
236)      GO TO 1601
237)  161  CONTINUE
238) C          FOR 1ST PASS
239)      DO 160 I=1,Y
240)      KONT=0
241)      DO 42 K=1,X
242)      EA(K)=0.0D0
243)      EB(K)=0.0D0
244)      C(K)=0.0D0
245)      D(K)=0.0D0
246)      O(K)=0.0D0
247)  42  CONTINUE
248)      DO 170 J=1,X
249)  1601  CONTINUE
250) C          FIGURE RESULTANT CONDUCTIVITIES BETWEEN NODES
251)      KONT=KONT+1
252)      INDEX=RAY(I,J,3)
253)      IF(I.EQ.1) GO TO 71
254)      TK1=(2.0D0*TK(I,J)*TK((I-1),J))/(TK(I,J)+TK((I-1),J))
255)  71  CONTINUE
256)      IF(J.EQ.1) GO TO 72
257)      TK2=(2.0D0*TK(I,J)*TK(I,(J-1)))/(TK(I,J)+TK(I,(J-1)))
258)  72  CONTINUE
259)      IF(I.EQ.Y) GO TO 73
260)      TK3=(2.0D0*TK(I,J)*TK((I+1),J))/(TK(I,J)+TK((I+1),J))
261)  73  CONTINUE
262)      IF(J.EQ.X) GO TO 74
263)      TK4=(2.0D0*TK(I,J)*TK(I,(J+1)))/(TK(I,J)+TK(I,(J+1)))
264)  74  CONTINUE
265) C          FIGURE R3 FOR TIME T-DELT/2, R4 FOR TIME T
266)      IF(ISTAT(I,J).NE.3) GO TO 75
267) C          NODE PRESENTLY PHASE CHANGE

```

```

268) R3=2.00*DS*DS*ROPC(INDEX)*CPPC(INDEX)/DELT
269) R4=R3
270) GO TO 78
271) 75 CONTINUE
272) C NODE NOT PHASE CHANGE PRESENTLY
273) R3=2.00*DS*DS*ROO(I,J)*CPO(I,J)/DELT
274) R4=2.00*DS*DS*RO(I,J)*CP(I,J)/DELT
275) 78 CONTINUE
276) C ROUTE CORNERS
277) IF(RAY(I,J,2).GE.5) GO TO 38
278) IJ2=RAY(I,J,2)
279) GO TO (101,102,103,104),IJ2
280) C ROUTE THE REST OF THE NODES
281) 38 CONTINUE
282) IJ2=RAY(I,J,2)-4
283) GO TO (105,106,107,108,109,110,111,112,113,114,115,
284) & 116,117,118,119,120,121,122,123,124,125,126,127,
285) & 128,129,130,131,132),IJ2
286) C
287) 110 CONTINUE
288) C SEMI-INFINITE BOUNDARY
289) C RIGHT SIDE
290) IF(J.NE.X) GO TO 1100
291) IFLAG(7)=11
292) IF(JFLG.NE.1) GO TO 1108
293) EA(KONT)=TK2/DI
294) EB(KONT)=-TK2/DI-DI2*R4
295) C(KONT)=0.00
296) D(KONT)=-TK1*DI2*RAY((I-1),J,1)-TK3*DI2*RAY((I+1),J,1)-
297) & (TK(I,J)/DI)*TMPR(I)+(TK1*DI2+TK3*DI2-DI2*R3+TK(I,J)/DI)
298) & *RAY(I,J,1)
299) GO TO 165
300) 1108 CONTINUE
301) EA(KONT)=TK1*DI2
302) EB(KONT)=-TK1*DI2-TK3*DI2-DI2*R4
303) C(KONT)=TK3*DI2
304) D(KONT)=-TK2/DI)*RAY(I,(J-1),1)-(TK(I,J)/DI)*TMPR(I)+
305) & (TK2/DI+TK(I,J)/DI-DI2*R3)*RAY(I,J,1)
306) GO TO 165
307) 1100 CONTINUE
308) C LEFT SIDE
309) IF(J.NE.1) GO TO 1101
310) IFLAG(3)=11
311) IF(JFLG.NE.1) GO TO 1104
312) EA(KONT)=0.00
313) EB(KONT)=-TK4/DI-DI2*R4
314) C(KONT)=TK4/DI
315) D(KONT)=-TK1*DI2*RAY((I-1),J,1)-TK3*DI2*RAY((I+1),J,1)-
316) & (TK(I,J)/DI)*TMPL(I)+(TK1*DI2+TK3*DI2-DI2*R3+TK(I,J)/DI)
317) & *RAY(I,J,1)
318) GO TO 165
319) 1104 CONTINUE
320) EA(KONT)=TK1*DI2
321) EB(KONT)=-TK1*DI2-TK3*DI2-DI2*R4
322) C(KONT)=TK3*DI2
323) D(KONT)=-TK4/DI)*RAY(I,(J+1),1)-(TK(I,J)/DI)*TMPL(I)+
324) & (TK4/DI+TK(I,J)/DI-DI2*R3)*RAY(I,J,1)
325) GO TO 165
326) 1101 CONTINUE
327) C TOP OF GRID
328) IF(I.NE.1) GO TO 1102
329) IFLAG(1)=11
330) WRITE(1,9997) I,J
331) GO TO 9999
332) 1102 CONTINUE
333) C BOTTOM OF GRID
334) IFLAG(5)=11
335) IF(JFLG.NE.1) GO TO 1106
336) EA(KONT)=TK2*DI2
337) EB(KONT)=-TK2*DI2-TK4*DI2-R4*DI2
338) C(KONT)=TK4*DI2
339) D(KONT)=-TK1/DI)*RAY((I-1),J,1)-(TK(I,J)/DI)*TMPB(J)+
340) & (TK1/DI+TK(I,J)/DI-R3*DI2)*RAY(I,J,1)
341) GO TO 165
342) 1106 CONTINUE
343) EA(KONT)=TK1/DI
344) EB(KONT)=-TK1/DI-R4*DI2
345) C(KONT)=0.00
346) D(KONT)=-TK2*DI2*RAY(I,(J-1),1)-TK4*DI2*RAY(I,(J+1),1)-
347) & (TK(I,J)/DI)*TMPB(J)+(TK2*DI2+TK4*DI2-DI2*R3+TK(I,J)/DI)
348) & *RAY(I,J,1)
349) GO TO 165
350) 109 CONTINUE
351) C CONVECTIVE SURFACE BOUNDARY
352) C TOP CONVECTIVE
353) IFLAG(1)=12
354) IF(JFLG.NE.1) GO TO 1092
355) EA(KONT)=.500*TK2
356) EB(KONT)=-.500*TK2-.500*TK4-.500*R4
357) C(KONT)=.500*TK4
358) D(KONT)=-TK3*RAY((I+1),J,1)+(TK3+H(INDEX)*DS-.500*R3)*
359) & RAY(I,J,1)-H(INDEX)*DS*TMPT(J)
360) GO TO 165
361) 1092 CONTINUE

```

```

362)      EA(KONT)=0.D0
363)      EB(KONT)=-TK3-.5D0*R4
364)      C(KONT)=TK3
365)      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)-.5D0*TK4*RAY(I,(J+1),1)+
366)      & (.5D0*TK2+.5D0*TK4-.5D0*R3+H(INDEX)*DS)*RAY(I,J,1)-H(INDEX)*DS*
367)      & TMPT(J)
368)      GO TO 165
369)      CONTINUE
132) C     RIGHT SIDE CONVECTIVE
370)      IFLAG(7)=12
371)      IF(JFLG.NE.1) GO TO 1094
372)      EA(KONT)=TK2
373)      EB(KONT)=-TK2-.5D0*R4
374)      C(KONT)=0.D0
375)      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-.5D0*TK3*RAY((I+1),J,1)+
376)      & (.5D0*TK1+.5D0*TK3-.5D0*R3+H(INDEX)*DS)*RAY(I,J,1)-
377)      & H(INDEX)*DS*TMPT(I)
378)      GO TO 165
379)      CONTINUE
1094) C    CONTINUE
380)      EA(KONT)=.5D0*TK1
381)      EB(KONT)=-.5D0*TK1-.5D0*TK3+.5D0*R4
382)      C(KONT)=.5D0*TK3
383)      D(KONT)=-TK2*RAY(I,(J-1),1)+(TK2-.5D0*R3+H(INDEX)*DS)*
384)      & RAY(I,J,1)-H(INDEX)*DS*TMPT(I)
385)      GO TO 165
386)      CONTINUE
131) C    BOTTOM CONVECTIVE
387)      IFLAG(5)=12
388)      IF(JFLG.NE.1) GO TO 1096
389)      EA(KONT)=.5D0*TK2
390)      EB(KONT)=-.5D0*TK2-.5D0*TK4-.5D0*R4
391)      C(KONT)=.5D0*TK4
392)      D(KONT)=-TK1*RAY((I-1),J,1)+(TK1+H(INDEX)*DS-.5D0*R3)*
393)      & RAY(I,J,1)-H(INDEX)*DS*TMPT(J)
394)      GO TO 165
395)      CONTINUE
1096) C    CONTINUE
396)      EA(KONT)=TK1
397)      EB(KONT)=-TK1-.5D0*R4
398)      C(KONT)=0.D0
399)      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)-.5D0*TK4*RAY(I,(J+1),1)+
400)      & (.5D0*TK2+.5D0*TK4-.5D0*R3+H(INDEX)*DS)*RAY(I,J,1)-H(INDEX)*DS*
401)      & TMPT(J)
402)      GO TO 165
403)      CONTINUE
130) C    LEFT SIDE CONVECTIVE
404)      IFLAG(3)=12
405)      IF(JFLG.NE.1) GO TO 1097
406)      EA(KONT)=0.D0
407)      EB(KONT)=-TK4-.5D0*R4
408)      C(KONT)=TK4
409)      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-.5D0*TK3*RAY((I+1),J,1)+
410)      & (.5D0*TK1+.5D0*TK3-.5D0*R3+H(INDEX)*DS)*RAY(I,J,1)-
411)      & H(INDEX)*DS*TMPL(I)
412)      GO TO 165
413)      CONTINUE
1097) C    CONTINUE
414)      EA(KONT)=.5D0*TK1
415)      EB(KONT)=-.5D0*TK1-.5D0*TK3+.5D0*R4
416)      C(KONT)=.5D0*TK3
417)      D(KONT)=-TK4*RAY(I,(J+1),1)+(TK4-.5D0*R3+H(INDEX)*DS)*
418)      & RAY(I,J,1)-H(INDEX)*DS*TMPL(I)
419)      GO TO 165
420)      CONTINUE
107) C    CONSTANT TEMP BOUNDARY
421)      EB(KONT)=1
422)      D(KONT)=RAY(I,J,1)
423)      IF(I.NE.1) GO TO 1070
424)      IFLAG(1)=10
425)      GO TO 165
426)      CONTINUE
1070) C   CONTINUE
427)      IF(J.NE.1) GO TO 1071
428)      IFLAG(3)=10
429)      GO TO 165
430)      CONTINUE
1071) C   CONTINUE
431)      IF(I.NE.Y) GO TO 1072
432)      IFLAG(5)=10
433)      GO TO 165
434)      CONTINUE
1072) C   CONTINUE
435)      IF(J.NE.X) GO TO 1073
436)      IFLAG(7)=10
437)      GO TO 165
438)      CONTINUE
1073) C   CONTINUE
439)      CONTINUE
106) C    CONSTANT INTERIOR NODE
440)      EB(KONT)=1
441)      D(KONT)=RAY(I,J,1)
442)      GO TO 165
443)      CONTINUE
105) C    VARIABLE INTERIOR NODE
444)      IF(JFLG.NE.1) GO TO 1050
445)      EA(KONT)=TK2
446)      C(KONT)=TK4
447)      EB(KONT)=-TK2-TK4-R4
448)      D(KONT)=-TK1*RAY((I-1),J,1)-TK3*RAY((I+1),J,1)+
449)      & (TK1+TK3-R3)*RAY(I,J,1)
450)
451)
452)
453)
454)
455)

```

```

456) GO TO 165
457) 1050 CONTINUE
458) EA(KONT)=TK1
459) C(KONT)=TK3
460) EB(KONT)=-TK1-TK3-R4
461) D(KONT)=-TK2*RAY(I,(J-1),1)-TK4*RAY(I,(J+1),1)+(TK2+TK4-R3)*
462) & RAY(I,J,1)
463) GO TO 165
464) 129 CONTINUE
465) C CONSTANT HEAT FLUX
466) C RIGHT SIDE
467) IFLAG(7)=13
468) IF(JFLG.NE.1) GO TO 1301
469) EA(KONT)=TK2
470) EB(KONT)=-TK2-.5D0*R4
471) D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-.5D0*TK3*RAY((I+1),J,1)+
472) & (.5D0*TK1+.5D0*TK3-.5D0*R3)*RAY(I,J,1)-FLXR(I)*DS
473) C(KONT)=0.D0
474) GO TO 165
475) 1301 CONTINUE
476) EA(KONT)=.5D0*TK1
477) EB(KONT)=-.5D0*TK1-.5D0*TK3-.5D0*R4
478) C(KONT)=.5D0*TK3
479) D(KONT)=-TK2*RAY(I,(J-1),1)+(TK2-.5D0*R3)*RAY(I,J,1)
480) & -FLXR(I)*DS
481) GO TO 165
482) 127 CONTINUE
483) C LEFT SIDE CONST FLUX
484) IFLAG(3)=13
485) IF(JFLG.NE.1) GO TO 1303
486) EA(KONT)=0.D0
487) EB(KONT)=-TK4-.5D0*R4
488) C(KONT)=TK4
489) D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-.5D0*TK3*RAY((I+1),J,1)+
490) & (.5D0*TK1+.5D0*TK3-.5D0*R3)*RAY(I,J,1)-FLXL(I)*DS
491) GO TO 165
492) 1303 CONTINUE
493) EA(KONT)=.5D0*TK1
494) EB(KONT)=-.5D0*TK1-.5D0*TK3-.5D0*R4
495) C(KONT)=.5D0*TK3
496) D(KONT)=-TK4*RAY(I,(J+1),1)+(TK4-.5D0*R3)*RAY(I,J,1)-FLXL(I)*DS
497) GO TO 165
498) 108 CONTINUE
499) C TOP CONST FLUX
500) IFLAG(1)=13
501) IF(JFLG.NE.1) GO TO 1305
502) EA(KONT)=.5D0*TK2
503) EB(KONT)=-.5D0*TK2-.5D0*TK4-.5D0*R4
504) C(KONT)=.5D0*TK4
505) D(KONT)=-TK3*RAY((I+1),J,1)+(TK3-.5D0*R3)*RAY(I,J,1)-FLXT(J)*DS
506) GO TO 165
507) 1305 CONTINUE
508) EA(KONT)=0.D0
509) EB(KONT)=-TK3-.5D0*R4
510) C(KONT)=TK3
511) D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)-.5D0*TK4*RAY(I,(J+1),1)+
512) & (.5D0*TK2+.5D0*TK4-.5D0*R3)*RAY(I,J,1)-FLXT(J)*DS
513) GO TO 165
514) 128 CONTINUE
515) C BOTTOM CONST FLUX
516) IF(JFLG.NE.1) GO TO 1307
517) IFLAG(5)=13
518) EA(KONT)=.5D0*TK2
519) EB(KONT)=-.5D0*TK2-.5D0*TK4-.5D0*R4
520) C(KONT)=.5D0*TK4
521) D(KONT)=-TK1*RAY((I-1),J,1)+(TK1-.5D0*R3)*RAY(I,J,1)
522) & -FLXB(J)*DS
523) GO TO 165
524) 1307 CONTINUE
525) EA(KONT)=TK1
526) EB(KONT)=-TK1-.5D0*R4
527) C(KONT)=0.D0
528) D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)-.5D0*TK4*RAY(I,(J+1),1)+
529) & (.5D0*TK2+.5D0*TK4-.5D0*R3)*RAY(I,J,1)
530) & -FLXB(J)*DS
531) GO TO 165
532) 111 CONTINUE
533) C NODE ADJ TO LEFT SEMI-INF
534) IF(JFLG.NE.1) GO TO 1110
535) EA(KONT)=TK2/DI
536) EB(KONT)=-TK4-TK2/DI-R4
537) C(KONT)=TK4
538) D(KONT)=-TK1*RAY((I-1),J,1)-TK3*RAY((I+1),J,1)+(TK1+
539) & TK3-R3)*RAY(I,J,1)
540) GO TO 165
541) 1110 CONTINUE
542) EA(KONT)=TK1
543) EB(KONT)=-TK1-TK3-R4
544) C(KONT)=TK3
545) D(KONT)=-TK4*RAY(I,(J+1),1)-(TK2/DI)*RAY(I,(J-1),1)+
546) & (TK4+TK2/DI-R3)*RAY(I,J,1)
547) GO TO 165
548) 112 CONTINUE
549) C NODE ADJ TO BOTTOM SEMI-INF

```



```

550) IF(JFLG.NE.1) GO TO 1120
551) EA(KONT)=TK2
552) EB(KONT)=-TK2-TK4-R4
553) C(KONT)=TK4
554) D(KONT)=-TK1*RAY((I-1),J,1)-(TK3/DI)*RAY((I+1),J,1)+
555) & (TK1+TK3/DI-R3)*RAY(I,J,1)
556) GO TO 165
557) 1120 CONTINUE
558) EA(KONT)=TK1
559) EB(KONT)=-TK1-TK3/DI-R4
560) C(KONT)=TK3/DI
561) D(KONT)=-TK2*RAY(I,(J-1),1)-TK4*RAY(I,(J+1),1)+(TK2+
562) & TK4-R3)*RAY(I,J,1)
563) GO TO 165
564) 113 CONTINUE
565) C NODE ADJ. TO RIGHT SEMI-INF
566) IF(JFLG.NE.1) GO TO 113C
567) EA(KONT)=TK2
568) EB(KONT)=-TK2-TK4/DI-R4
569) C(KONT)=TK4/DI
570) D(KONT)=-TK1*RAY((I-1),J,1)-TK3*RAY((I+1),J,1)+
571) & (TK1+TK3-R3)*RAY(I,J,1)
572) GO TO 165
573) 113C CONTINUE
574) EA(KONT)=TK1
575) EB(KONT)=-TK1-TK3-R4
576) C(KONT)=TK3
577) D(KONT)=-TK4/DI)*RAY(I,(J+1),1)-TK2*RAY(I,(J-1),1)+
578) & (TK2+TK4/DI-R3)*RAY(I,J,1)
579) GO TO 165
580) 115 CONTINUE
581) C SQUARE NODE ADJ TO 2 SEMI-INF SIDES
582) IF(J.NE.2) GO TO 1150
583) IF(I.NE.2) GO TO 1151
584) C TOP LEFT
585) WRITE(1,9997) I,J
586) GO TO 9999
587) 1151 CONTINUE
588) C BOTTOM LEFT
589) IF(JFLG.NE.1) GO TO 1152
590) EA(KONT)=TK2/DI
591) EB(KONT)=-TK2/DI-TK4-R4
592) C(KONT)=TK4
593) D(KONT)=-TK1*RAY((I-1),J,1)+(TK3/DI)*RAY((I+1),J,1)+(TK1+
594) & TK3/DI-R3)*RAY(I,J,1)
595) GO TO 165
596) 1152 CONTINUE
597) EA(KONT)=TK1
598) EB(KONT)=-TK1-TK3/DI-R4
599) C(KONT)=TK3/DI
600) D(KONT)=-TK2/DI)*RAY(I,(J-1),1)-TK4*RAY(I,(J+1),1)+
601) & (TK2/DI+TK4-R3)*RAY(I,J,1)
602) GO TO 165
603) 1150 CONTINUE
604) IF(I.NE.2) GO TO 1153
605) C TOP RIGHT
606) WRITE(1,9997) I,J
607) GO TO 9999
608) 1153 CONTINUE
609) C BOTTOM RIGHT
610) IF(JFLG.NE.1) GO TO 1154
611) EA(KONT)=TK2
612) EB(KONT)=-TK2-TK4/DI-R4
613) C(KONT)=TK4/DI
614) D(KONT)=-TK1*RAY((I-1),J,1)+(TK3/DI)*RAY((I+1),J,1)+
615) & (TK1+TK3/DI-R3)*RAY(I,J,1)
616) GO TO 165
617) 1154 CONTINUE
618) EA(KONT)=TK1
619) EB(KONT)=-TK1-TK3/DI-R4
620) C(KONT)=TK3/DI
621) D(KONT)=-TK2*RAY(I,(J-1),1)-(TK4/DI)*RAY(I,(J+1),1)+
622) & (TK2+TK4/DI-R3)*RAY(I,J,1)
623) GO TO 165
624) 121 CONTINUE
625) C RIGHT SIDE CONST FLUX ADJ. TO BOTOM SEMI-INF
626) IF(JFLG.NE.1) GO TO 1210
627) EA(KONT)=TK2
628) EB(KONT)=-TK2-.5D0*R4
629) C(KONT)=0.D0
630) D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-(TK3/(2.D0*DI))*RAY((I+1),J,1)-
631) & FLXR(I)*DS+(.5D0*TK1+TK3/(2.D0*DI)-.5D0*R3)*RAY(I,J,1)
632) GO TO 165
633) 1210 CONTINUE
634) EA(KONT)=.5D0*TK1
635) EB(KONT)=-.5D0*TK1-TK3/(2.D0*DI)-.5D0*R4
636) C(KONT)=TK3/(2.D0*DI)
637) D(KONT)=-TK2*RAY(I,(J-1),1)-FLXR(I)*DS+(TK2-.5D0*R3)*RAY(I,J,1)
638) GO TO 165
639) 123 CONTINUE
640) C LEFT SIDE CONST FLUX ADJ TO BOTOM SEMI-INF

```

```

641) IF(JFLG.NE.1) GO TO 1230
642) EA(KONT)=0.D0
643) EB(KONT)=-TK4-.5D0*R4
644) C(KONT)=TK4
645) D(KONT)=-.5D0*TK1+RAY((I-1),J,1)-(TK3/(2.D0*DI))*RAY((I+1),J,1)-
646) & FLXL(I)*DS+(.5D0*TK1+TK3/(2.D0*DI)-.5D0*R3)*RAY(I,J,1)
647) GO TO 165
648) 1230 CONTINUE
649) EA(KONT)=.5D0*TK1
650) EB(KONT)=-.5D0*TK1-TK3/(2.D0*DI)-.5D0*R4
651) C(KONT)=TK3/(2.D0*DI)
652) D(KONT)=-TK4*RAY(I,(J+1),1)-FLXL(I)*DS+(TK4-.5D0*R3)*RAY(I,J,1)
653) GO TO 165
654) 116 CONTINUE
655) C SEMI-INF NODE ABOVE SEMI-INF CORNER NODE
656) IF(J.NE.1) GO TO 1165
657) C BOTTOM LEFT
658) IF(JFLG.NE.1) GO TO 1160
659) EA(KONT)=0.0D0
660) EB(KONT)=-TK(I,J)/DI-TK4/DI+R4*DI2
661) C(KONT)=TK4/DI
662) D(KONT)=-TK1*DI2*RAY((I-1),J,1)-(TK3*DI2/DI)*RAY((I+1),J,1)
663) & -(TK(I,J)/DI)*TMPL(I)+(TK1*DI2+TK3*DI2/DI-R3*DI2)*RAY(I,J,1)
664) GO TO 165
665) 1160 CONTINUE
666) EA(KONT)=TK1*DI2
667) EB(KONT)=-TK1+TK3/DI-R4)*DI2
668) C(KONT)=TK3*DI2/DI
669) D(KONT)=-TK(I,J)/DI)*TMPL(I)-(TK4/DI)*RAY(I,(J+1),1)+
670) & (TK(I,J)/DI+TK4/DI-R3*DI2)*RAY(I,J,1)
671) GO TO 165
672) 1165 CONTINUE
673) C BOTTOM RIGHT
674) IF(JFLG.NE.1) GO TO 1166
675) EA(KONT)=TK2/DI
676) EB(KONT)=-TK(I,J)/DI-TK2/DI+R4*DI2
677) C(KONT)=0.0D0
678) D(KONT)=-TK1*DI2*RAY((I-1),J,1)-(TK3*DI2/DI)*RAY((I+1),J,1)
679) & -(TK(I,J)/DI)*TMPL(I)+(TK1*DI2+TK3*DI2/DI-R3*DI2)*RAY(I,J,1)
680) GO TO 165
681) 1166 CONTINUE
682) EA(KONT)=TK1*DI2
683) EB(KONT)=-TK1+TK3/DI-R4)*DI2
684) C(KONT)=TK3*DI2/DI
685) D(KONT)=-TK(I,J)/DI)*TMPL(I)-(TK2/DI)*RAY(I,(J-1),1)+
686) & (TK(I,J)/DI+TK2/DI-R3*DI2)*RAY(I,J,1)
687) GO TO 165
688) 117 CONTINUE
689) C SEMI-INF NODE TO LEFT OF SEMI-INF CORNER NODE
690) IF(JFLG.NE.1) GO TO 1170
691) EA(KONT)=TK2*DI2
692) EB(KONT)=-TK4/DI-TK2-R4)*DI2
693) C(KONT)=TK4*DI2/DI
694) D(KONT)=-TK1/DI)*RAY((I-1),J,1)-(TK(I,J)/DI)*TMPB(J)
695) & +(TK1/DI+TK(I,J)/DI-R3*DI2)*RAY(I,J,1)
696) GO TO 165
697) 1170 CONTINUE
698) EA(KONT)=TK1/DI
699) EB(KONT)=-TK1+TK(I,J)/DI-R4)*DI2
700) C(KONT)=0.0D0
701) D(KONT)=-TK4*DI2/DI)*RAY(I,(J+1),1)-TK2*DI2*RAY(I,(J-1),1)-
702) & (TK(I,J)/DI)*TMPB(J)+(TK2/DI+TK4-R3)*DI2*RAY(I,J,1)
703) GO TO 165
704) 119 CONTINUE
705) C SEMI-INF NODE TO RIGHT OF SEMI-INF CORNER NODE
706) IF(JFLG.NE.1) GO TO 1190
707) EA(KONT)=TK2*DI2/DI
708) EB(KONT)=-TK2/DI-TK4-R4)*DI2
709) C(KONT)=TK4*DI2
710) D(KONT)=-TK1/DI)*RAY((I-1),J,1)-(TK(I,J)/DI)*TMPB(J)+
711) & (TK1/DI+TK(I,J)/DI-R3*DI2)*RAY(I,J,1)
712) GO TO 165
713) 1190 CONTINUE
714) EA(KONT)=TK1/DI
715) EB(KONT)=-TK1+TK(I,J)/DI-R4)*DI2
716) C(KONT)=0.0D0
717) D(KONT)=-TK2*DI2/DI)*RAY(I,(J-1),1)-TK4*DI2*RAY(I,(J+1),1)-
718) & (TK(I,J)/DI)*TMPB(J)+(TK2/DI+TK4-R3)*DI2*RAY(I,J,1)
719) GO TO 165
720) 120 CONTINUE
721) C TOP CONST FLUX NODE ADJ TO RIGHT SEMI-INF
722) IF(JFLG.NE.1) GO TO 1200
723) EA(KONT)=.50D0*TK2
724) EB(KONT)=-.50D0*(TK2+TK4/DI+R4)
725) C(KONT)=TK4/(2.0D0*DI)
726) D(KONT)=-TK3*RAY((I+1),J,1)-DS*FLXT(J)+(TK3-.50D0*R3)
727) & *RAY(I,J,1)
728) GO TO 165
729) 1200 CONTINUE
730) EA(KONT)=0.0D0
731) EB(KONT)=-TK3-.50D0*R4
732) C(KONT)=TK3
733) D(KONT)=-1.*(TK4/(2.0D0*DI))*RAY(I,(J+1),1)-.5*TK2

```

```

734)      & *RAY(I,(J-1),1)+(TK4/(2.0D0*DI)+.50D0*TK2-.50D0*R3)
735)      & *RAY(I,J,1)-DS*FLXT(J)
736)      122  CONTINUE
737)      C      TOP CONST FLUX ADJAC TO LEFT SEMI-INF
738)      IF(JFLG.NE.1) GO TO 1220
739)      EA(KONT)=TK2/(2.0D0*DI)
740)      EB(KONT)=-.50D0*(TK2/DI+TK4+R4)
741)      C(KONT)=-.50D0*TK4
742)      D(KONT)=-TK3*RAY((I+1),J,1)-DS*FLXT(J)+
743)      & (TK3-.50D0*R3)*RAY(I,J,1)
744)      GO TO 165
745)      1220  CONTINUE
746)      EA(KONT)=0.0D0
747)      EB(KONT)=-TK3-.5*R4
748)      C(KONT)=TK3
749)      D(KONT)=-TK2/(2.0D0*DI)*RAY(I,(J-1),1)-.50D0*TK4*
750)      & RAY(I,(J+1),1)+(TK2/(2.0D0*DI)+.50D0*TK4-.50D0*R3)
751)      & *RAY(I,J,1)-DS*FLXT(J)
752)      GO TO 165
753)      114  CONTINUE
754)      118  CONTINUE
755)      124  CONTINUE
756)      125  CONTINUE
757)      126  CONTINUE
758)      WRITE(1,9997) I,J
759)      GO TO 9999
760)      C      (((((((((((((((((((((((((((((((
761)      C      EQUATIONS FOR THE CORNERS OF THE GRID
762)      101  CONTINUE
763)      C      TOP LEFT-HAND CORNER
764)      GO TO (201,202,203,204,205,206,207,208,209,210),KRRR(1)
765)      202  CONTINUE
766)      C      CONSTANT FLUX ON BOTH SIDES
767)      IFLAG(2)=1
768)      IF(JFLG.NE.1) GO TO 2020
769)      EA(KONT)=0.0D0
770)      EB(KONT)=-.50D0*TK4-.25D0*R4
771)      C(KONT)=-.50D0*TK4
772)      D(KONT)=-.50D0*TK3*RAY((I+1),J,1)+(.50D0*TK3-.25D0*R3)*RAY(I,J,1)-
773)      & .50D0*DS*(FLXT(J)+FLXL(I))
774)      GO TO 165
775)      2020  CONTINUE
776)      EA(KONT)=0.0D0
777)      EB(KONT)=-.50D0*TK3-.25D0*R4
778)      C(KONT)=-.50D0*TK3
779)      D(KONT)=-.50D0*TK4*RAY(I,(J+1),1)+(.50D0*TK4-.25D0*R3)*
780)      & RAY(I,J,1)-.50D0*DS*(FLXT(J)+FLXL(I))
781)      GO TO 165
782)      203  CONTINUE
783)      C      CONVECTIVE ON BOTH SIDES
784)      IFLAG(2)=2
785)      IF(JFLG.NE.1) GO TO 2030
786)      EA(KONT)=0.0D0
787)      EB(KONT)=-.50D0*TK4-.25D0*R4
788)      C(KONT)=-.50D0*TK4
789)      D(KONT)=-.50D0*TK3*RAY((I+1),J,1)+(.50D0*TK3+H(INDEX)*DS-.25D0*R3)*
790)      & RAY(I,J,1)-.50D0*H(INDEX)*TMPT(J)-.50D0*H(INDEX)*TMPL(I)
791)      GO TO 165
792)      2030  CONTINUE
793)      EA(KONT)=0.0D0
794)      EB(KONT)=-.50D0*TK3-.25D0*R4
795)      C(KONT)=-.50D0*TK3
796)      D(KONT)=-.50D0*TK4*RAY(I,(J+1),1)+(.50D0*TK4+H(INDEX)*DS-.25D0*R3)*
797)      & RAY(I,J,1)-.50D0*H(INDEX)*(TMPT(J)+TMPL(I))
798)      GO TO 165
799)      204  CONTINUE
800)      C      SEMI-INF ON BOTH SIDES
801)      IFLAG(2)=3
802)      WRITE(1,9997) I,J
803)      GO TO 9999
804)      206  CONTINUE
805)      C      VERT CONST FLUX--HORIZ CONVECT
806)      IFLAG(2)=4
807)      IF(JFLG.NE.1) GO TO 2060
808)      EA(KONT)=0.0D0
809)      EB(KONT)=-.50D0*TK4-.25D0*R4
810)      C(KONT)=-.50D0*TK4
811)      D(KONT)=-.50D0*TK3*RAY((I+1),J,1)-.50D0*FLXL(I)*DS
812)      & -.50D0*H(INDEX)*DS*TMPT(J)+(.50D0*TK3+.50D0*H(INDEX)*DS
813)      & -.25D0*R3)*RAY(I,J,1)
814)      GO TO 165
815)      2060  CONTINUE
816)      EA(KONT)=0.0D0
817)      EB(KONT)=-.50D0*TK3-.50D0*H(INDEX)*DS
818)      C(KONT)=-.50D0*TK3
819)      D(KONT)=-.50D0*TK4*RAY(I,(J+1),1)-.50D0*FLXL(I)*DS
820)      & -.50D0*H(INDEX)*DS*TMPT(J)+(.50D0*TK4-.25D0*R3)
821)      & *RAY(I,J,1)
822)      GO TO 165
823)      205  CONTINUE
824)      C      VERT CONVECT--HORIZ CONST FLUX
825)      IFLAG(2)=7
826)      IF(JFLG.NE.1) GO TO 2050
827)      EA(KONT)=0.0D0

```

```

828) EB(KONT)=-.5000*TK4-.5000*H(INDEX)*DS-.25000*R4
829) C(KONT)=-.5000*TK4
830) D(KONT)=-.5000*FLXT(J)*DS-.5000*TK3*RAY((I+1),J,1)+
831) & (.5000*TK3-.25000*R3)*RAY(I,J,1)-.5000*H(INDEX)*DS*TMPL(I)
832) GO TO 165
833) 2050 CONTINUE
834) EA(KONT)=0.000
835) EB(KONT)=-.5000*TK3-.25000*R4
836) C(KONT)=-.5000*TK3
837) D(KONT)=-.5000*TK4*RAY(I,(J+1),1)-.5000*H(INDEX)*DS*TMPL(I)
838) & +(.5000*TK4+.5000*H(INDEX)*DS-.5000*FLXT(J)*DS-.25000*R3)*
839) & RAY(I,J,1)
840) GO TO 165
841)
842) 208 CONTINUE
843) C VERT=SEMI-INF,HORIZ=CONST FLUX
844) IF(JFLG.NE.1) GO TO 2080
845) EA(KONT)=0.000
846) EB(KONT)=-TK(I,J)/(2.000*DI)+TK4/(2.000*DI)
847) & +.5000*DI2*R4)
848) C(KONT)=TK4/(2.000*DI)
849) D(KONT)=-FLXT(J)*DI2*DS-TK3*DI2*RAY((I+1),J,1)+
850) & (TK3*DI2-.5000*DI2*R3)*RAY(I,J,1)-(TK(I,J)/(2.000*DI))*TMPL(I)
851) GO TO 165
852) 2080 CONTINUE
853) EA(KONT)=0.000
854) EB(KONT)=-TK3+.5000*R4)*DI2
855) C(KONT)=TK3*DI2
856) D(KONT)=-FLXT(J)*DI2*DS-(TK(I,J)/(2.000*DI))*TMPL(I)
857) & -(TK4/(2.000*DI))*RAY(I,(J+1),1)+(TK(I,J)/(2.000*DI)+
858) & TK4/(2.000*DI)-.5000*DI2*R3)*RAY(I,J,1)
859) GO TO 165
860) 210 CONTINUE
861) C VERT=SEMI-INF,HORIZ=CONVECT
862) IFLAG(2)=8
863) IF(JFLG.NE.1) GO TO 2100
864) EA(KONT)=0.000
865) EB(KONT)=-TK4+TK(I,J)/(2.000*DI)-.5000*DI2*R4
866) C(KONT)=TK4/(2.000*DI)
867) D(KONT)=-H(INDEX)*DI2*TMPL(J)-(TK(I,J)/(2.000*DI))*TMPL(I)
868) & -TK3*DI2*RAY((I+1),J,1)+(TK3*DI2+H(INDEX)*DI2-.5000*DI2*R3)
869) & *RAY(I,J,1)
870) GO TO 165
871) 2100 CONTINUE
872) EA(KONT)=0.000
873) EB(KONT)=-TK3*DI2-H(INDEX)*DI2-.5000*DI2*R4
874) C(KONT)=TK3*DI2
875) D(KONT)=-H(INDEX)*DI2*TMPL(J)-(TK4/(2.000*DI))*RAY(I,(J+1),1)
876) & -(TK(I,J)/(2.000*DI))*TMPL(I)+((TK4+TK(I,J))/(2.000*DI)-
877) & .5000*DI2*R3)*RAY(I,J,1)
878) GO TO 165
879) 207 CONTINUE
880) 209 CONTINUE
881) WRITE(1,9997) I,J
882) GO TO 9999
883) 201 CONTINUE
884) C CONSTANT CORNER NODE
885) EB(KONT)=1
886) D(KONT)=RAY(I,J,1)
887) IFLAG(2)=10
888) GO TO 165
889) 102 CONTINUE
890) C BOTTOM LEFT-HAND CORNER
891) GO TO (211,212,213,214,215,216,217,218,219,220),KRRR(2)
892) 212 CONTINUE
893) C CONST FLUX ON BOTH SIDES
894) IFLAG(4)=1
895) IF(JFLG.NE.1) GO TO 2120
896) EA(KONT)=0.000
897) EB(KONT)=-.5000*TK4-.25000*R4
898) C(KONT)=-.5000*TK4
899) D(KONT)=-.5000*TK1*RAY((I-1),J,1)+(.5000*TK1-.25000*R3)*RAY(I,J,1)
900) & -.5*DS*(FLXB(J)+FLXL(I))
901) GO TO 165
902) 2120 CONTINUE
903) EA(KONT)=-.5000*TK1
904) EB(KONT)=-.5000*TK1-.25000*R4
905) C(KONT)=0.000
906) D(KONT)=-.5000*TK4*RAY(I,(J+1),1)+(.5000*TK4-.25000*R3)*RAY(I,J,1)
907) & -.5*DS*(FLXB(J)+FLXL(I))
908) GO TO 165
909) 213 CONTINUE
910) C CONVECT ON BOTH SIDES
911) IFLAG(4)=2
912) IF(JFLG.NE.1) GO TO 2130
913) EA(KONT)=0.000
914) EB(KONT)=-.5000*TK4-.25000*R4
915) C(KONT)=-.5000*TK4
916) D(KONT)=-.5000*TK1*RAY((I-1),J,1)+(.5000*TK1+H(INDEX)*DS-.25000*R3)*
917) & RAY(I,J,1)-.5000*H(INDEX)*(TMPL(I)+TMPB(J))
918) GO TO 165
919) 2130 CONTINUE
920) EA(KONT)=-.5000*TK1
921) EB(KONT)=-.5000*TK1-.25000*R4

```

```

922) C(KONT)=0.00
923) D(KONT)=-.500*TK4*RAY(I,(J+1),1)+(.500*TK4+H(INDEX)*DS-.2500*R3)*
924) & RAY(I,J,1)-.500*H(INDEX)*(TMPL(I)+TMPB(J))
925) GO TO 165
926) CONTINUE
927) C 214 SEMI-INF ON BOTH SIDES
928) IFLAG(4)=3
929) DINF=DI2/DI
930) IF(JFLG.NE.1) GO TO 2140
931) EA(KONT)=0.00
932) EB(KONT)=-TK4*DINF-(DI2*DI2)*R4
933) C(KONT)=TK4*DINF
934) D(KONT)=-TK1*DINF*RAY((I-1),J,1)-TK(I,J)*(DINF)*
935) & TMPL(I)-TK(I,J)*DINF*TMPB(J)+(TK1*DINF
936) & +2.00*TK(I,J)*DINF-(DI2*DI2)*R3)*RAY(I,J,1)
937) GO TO 165
938) 2140 CONTINUE
939) EA(KONT)=TK1*DINF
940) EB(KONT)=-TK1*DINF-(DI2*DI2)*R4
941) C(KONT)=0.00
942) D(KONT)=-TK(I,J)*DINF*TMPL(I)-TK4*(DINF)*
943) & RAY(I,(J+1),1)+(2.00*TK(I,J)*DINF+TK4*DINF-
944) & (DI2*DI2)*R3)*RAY(I,J,1)-(TK(I,J)*DINF)*TMPB(J)
945) GO TO 165
946) 216 CONTINUE
947) C VERT CONST FLUX--HORIZ CONVECT
948) IFLAG(4)=4
949) IF(JFLG.NE.1) GO TO 2160
950) EA(KONT)=0.000
951) EB(KONT)=-.5000*TK4-.25000*R4
952) C(KONT)=.5000*TK4
953) D(KONT)=-.5000*TK1*RAY((I-1),J,1)-.5000*FLXL(I)*DS-.5000*
954) & H(INDEX)*DS*TMPB(J)+(.5000*TK3+.5000*H(INDEX)*DS-.25000*R3)*
955) & RAY(I,J,1)
956) GO TO 165
957) 2160 CONTINUE
958) EA(KONT)=.5000*TK1
959) EB(KONT)=-.5000*TK1-.5000*H(INDEX)*DS
960) C(KONT)=0.000
961) D(KONT)=-.5000*TK4*RAY(I,(J+1),1)-.5000*FLXL(I)*DS-.5000*H(INDEX)*
962) & DS*TMPB(J)+(.5000*TK4-.25000*R3)*RAY(I,J,1)
963) GO TO 165
964) 215 CONTINUE
965) C VERT CONVECT--HORIZ CONST FLUX
966) IFLAG(4)=7
967) IF(JFLG.NE.1) GO TO 2150
968) EA(KONT)=0.000
969) EB(KONT)=-.5000*TK4-.5000*H(INDEX)*DS-.25000*R4
970) C(KONT)=-.5000*TK4
971) D(KONT)=-.25000*H(INDEX)*DS*TMPL(I)-.5000*FLXB(J)*DS-.5000*TK1
972) & *RAY((I-1),J,1)+(.5000*TK1-.25000*R3)*RAY(I,J,1)
973) GO TO 165
974) 2150 CONTINUE
975) EA(KONT)=.5000*TK1
976) EB(KONT)=-.5000*TK1-.25000*R4
977) C(KONT)=0.000
978) D(KONT)=-.5000*TK4*RAY(I,(J+1),1)-.5000*H(INDEX)*DS*TMPL(I)
979) & +(.5000*TK4+.5000*H(INDEX)*DS-.5000*FLXB(J)*DS-.25000*R3)
980) & *RAY(I,J,1)
981) GO TO 165
982) 217 CONTINUE
983) C VERT CONST FLUX--HORIZ SEMI-INF
984) IFLAG(4)=5
985) IF(JFLG.NE.1) GO TO 2170
986) EA(KONT)=0.00
987) EB(KONT)=-TK4*DI2-.2500*DI2*R4
988) C(KONT)=TK4*DI2
989) D(KONT)=-TK1/(2.00*DI)*RAY((I-1),J,1)-(TK(I,J)/(2.00*DI))*
990) & TMPB(J)-FLXL(I)*DI2*DS+(TK1/(2.00*DI)+TK(I,J)/(2.00*DI)-
991) & .2500*DI2*R3)*RAY(I,J,1)
992) GO TO 165
993) 2170 CONTINUE
994) EA(KONT)=TK1/(2.00*DI)
995) EB(KONT)=-TK1/(2.00*DI)-TK(I,J)/(2.00*DI)-.2500*DI2*R4
996) C(KONT)=0.00
997) D(KONT)=-TK4*DI2*RAY(I,(J+1),1)-(TK(I,J)/(2.00*DI))*TMPB(J)-
998) & FLXL(I)*DI2*DS+(TK4*DI2-.2500*DI2*R3)*RAY(I,J,1)
999) GO TO 165
1000) 218 CONTINUE
1001) C VERT=SEMI-INF,HORIZ=CONST FLUX
1002) IFLAG(4)=9
1003) IF(JFLG.NE.1) GO TO 2180
1004) EA(KONT)=0.000
1005) EB(KONT)=-TK4+TK(I,J)/(2.000*DI)-.5000*DI2*R4
1006) C(KONT)=TK4/(2.000*DI)
1007) D(KONT)=-FLXB(J)*DI2*DS-(TK(I,J)/(2.000*DI))*TMPL(I)-TK1*DI2
1008) & *RAY((I-1),J,1)+(TK1*DI2-.5000*DI2*R3)*RAY(I,J,1)
1009) GO TO 165
1010) 2180 CONTINUE
1011) EA(KONT)=TK1*DI2
1012) EB(KONT)=-TK1*DI2-.5000*DI2*R4
1013) C(KONT)=0.000
1014) D(KONT)=-TK4/(2.000*DI)*RAY(I,(J+1),1)-(TK(I,J)/(2.000*DI))*

```

```

1015) & TMPL(I)-FLXB(J)*DI2*DS+((TK4+TK(I,J))/(2.000*DI)-.5000*DI2*R3)
1016) & *RAY(I,J,1)
1017) GO TO 165
1018) 219 CONTINUE
1019) C VERT=CONVECT,HORIZ=SEMI-INF
1020) IFLAG(4)=6
1021) IF(JFLG.NE.1) GO TO 2190
1022) EA(KONT)=0.000
1023) EB(KONT)=-TK4*DI2-H(INDEX)*DI2-.5000*DI2*R4
1024) C(KONT)=TK4*DI2
1025) D(KONT)=-H(INDEX)*DI2*TMPL(I)-(TK1/(2.000*DI))*RAY((I-1),J,1)
1026) & -(TK(I,J)/(2.000*DI))*TMPB(J)+((TK1+TK(I,J))/(2.000*DI)
1027) & -.5000*DI2*R3)*RAY(I,J,1)
1028) GO TO 165
1029) 2190 CONTINUE
1030) EA(KONT)=TK1/(2.000*DI)
1031) EB(KONT)=-TK1+TK(I,J)/(2.000*DI)-.5000*DI2*R4
1032) C(KONT)=0.000
1033) D(KONT)=-H(INDEX)*DI2*TMPL(I)-(TK(I,J)/(2.000*DI))*TMPB(J)
1034) & -TK4*DI2*RAY(I,(J+1),1)+(TK4*DI2+H(INDEX)*DI2-.5000*DI2*R3)
1035) & *RAY(I,J,1)
1036) GO TO 165
1037) 220 CONTINUE
1038) C VERT=SEMI-INF,HORIZ=CONVECT
1039) IFLAG(4)=8
1040) IF(JFLG.NE.1) GO TO 2200
1041) EA(KONT)=0.000
1042) EB(KONT)=-TK4+TK(I,J)/(2.000*DI)-.5000*DI2*R4
1043) C(KONT)=TK4/(2.000*DI)
1044) D(KONT)=-H(INDEX)*DI2*TMPB(J)-(TK(I,J)/(2.000*DI))*TMPL(I)
1045) & -TK1*DI2*RAY((I-1),J,1)+(TK1*DI2+H(INDEX)*DI2-.5000*DI2*R3)
1046) & *RAY(I,J,1)
1047) GO TO 165
1048) 2200 CONTINUE
1049) EA(KONT)=TK1*DI2
1050) EB(KONT)=-TK1*DI2-H(INDEX)*DI2-.5000*DI2*R4
1051) C(KONT)=0.000
1052) D(KONT)=-H(INDEX)*DI2*TMPB(J)-(TK4/(2.000*DI))*RAY(I,(J+1),1)
1053) & -(TK(I,J)/(2.000*DI))*TMPL(I)+((TK4+TK(I,J))/(2.000*DI)
1054) & -.5000*DI2*R3)*RAY(I,J,1)
1055) GO TO 165
1056) 211 CONTINUE
1057) C CONST CORNER NODE
1058) EB(KONT)=1
1059) D(KONT)=RAY(I,J,1)
1060) IFLAG(4)=10
1061) GO TO 165
1062) 103 CONTINUE
1063) C BOTTOM RIGHT-HAND CORNER
1064) GO TO (221,222,223,224,225,226,227,228,229,230),KRRN(3)
1065) 222 CONTINUE
1066) C CONST FLUX ON BOTH SIDES
1067) IF(JFLG.NE.1) GO TO 2220
1068) EA(KONT)=.500*TK2
1069) EB(KONT)=-.500*TK2-.2500*R4
1070) C(KONT)=0.00
1071) D(KONT)=-.500*TK1*RAY((I-1),J,1)+(.500*TK1-.2500*R3)*RAY(I,J,1)
1072) & -.500*DS*(FLXB(J)+FLXR(I))
1073) GO TO 165
1074) 2220 CONTINUE
1075) EA(KONT)=.500*TK1
1076) EB(KONT)=-.500*TK1-.2500*R4
1077) C(KONT)=0.00
1078) D(KONT)=-.500*TK2*RAY(I,(J-1),1)+(.500*TK2-.2500*R3)*RAY(I,J,1)
1079) & -.500*DS*(FLXB(J)+FLXR(I))
1080) IFLAG(6)=1
1081) GO TO 165
1082) 223 CONTINUE
1083) C CONVECTIVE ON BOTH SIDES
1084) IFLAG(6)=2
1085) IF(JFLG.NE.1) GO TO 2230
1086) EA(KONT)=.500*TK2
1087) EB(KONT)=-.500*TK2-.2500*R4
1088) C(KONT)=0.00
1089) D(KONT)=-.500*TK1*RAY((I-1),J,1)+(.500*TK1+H(INDEX)*DS-.2500*R3)*
1090) & RAY(I,J,1)-.500*H(INDEX)*(TMPB(J)+TMPR(I))
1091) GO TO 165
1092) 2230 CONTINUE
1093) EA(KONT)=.500*TK1
1094) EB(KONT)=-.500*TK1-.2500*R4
1095) D(KONT)=0.00
1096) D(KONT)=-.500*TK2*RAY(I,(J-1),1)+(.500*TK2+H(INDEX)*DS-.2500*R3)*
1097) & RAY(I,J,1)-.500*H(INDEX)*(TMPR(I)+TMPB(J))
1098) GO TO 165
1099) 224 CONTINUE
1100) C SEMI-INF ON BOTH SIDES
1101) IFLAG(6)=3
1102) DINF=DI2/DI
1103) IF(JFLG.NE.1) GO TO 2240
1104) EA(KONT)=TK2*DINF
1105) EB(KONT)=-TK2*DINF-(DI2*DI2)*R4
1106) C(KONT)=0.00
1107) D(KONT)=-TK1*DINF*RAY((I-1),J,1)-TK(I,J)*DINF*

```

```

1108) & TMPR(I)-TK(I,J)*DINF*TMPB(J)+(TK1*DINF+
1109) & 2.00*TK(I,J)*DINF-(DI2*DI2)*R3)*RAY(I,J,1)
1110) GO TO 165
1111) 2240 CONTINUE
1112) EA(KONT)=TK1*DINF
1113) EB(KONT)=-TK1*DINF-(DI2*DI2)*R4
1114) C(KONT)=0.00
1115) D(KONT)=-TK(I,J)*DINF*TMPR(I)-TK2*DINF*
& RAY(I,(J-1),1)+(2.00*TK(I,J)*DINF+TK2*DINF-
1116) & DI2*DI2*R3)*RAY(I,J,1)-(TK(I,J)*DINF)*TMPB(J)
1117) GO TO 165
1118) 226 CONTINUE
1119) C VERT CONST FLUX-- HORIZ CONVECT
1120) IFLAG(6)=4
1121) IF(JFLG.NE.1) GO TO 2260
1122) EA(KONT)=.5000*TK2
1123) EB(KONT)=-.5000*TK2-.2500*R4
1124) C(KONT)=0.000
1125) D(KONT)=-.5000*TK1*RAY((I-1),J,1)-.5000*FLXR(I)*DS-.5000*H(INDEX)
& *DS*TMPB(J)+(.5000*TK1+.5000*H(INDEX)*DS-.2500*R3)*RAY(I,J,1)
1126) GO TO 165
1127) 2260 CONTINUE
1128) EA(KONT)=.5000*TK1
1129) EB(KONT)=-.5000*TK1-.5000*H(INDEX)*DS
1130) C(KONT)=0.000
1131) D(KONT)=-.5000*TK2*RAY(I,(J-1),1)-.5000*FLXR(I)*DS-.5000*H(INDEX)
& *DS*TMPB(J)+(.5000*TK2-.2500*R3)*RAY(I,J,1)
1132) GO TO 165
1133) 225 CONTINUE
1134) C VERT CONVECT-- HORIZ CONST FLUX
1135) IFLAG(6)=7
1136) IF(JFLG.NE.1) GO TO 2250
1137) EA(KONT)=.5000*TK2
1138) EB(KONT)=-.5000*TK2-.5000*H(INDEX)*DS-.2500*R4
1139) C(KONT)=0.000
1140) D(KONT)=-.5000*H(INDEX)*DS*TMPR(I)-.5000*FLXB(J)*DS-.5000*TK1
& *RAY((I-1),J,1)+(.5000*TK1-.2500*R3)*RAY(I,J,1)
1141) GO TO 165
1142) 2250 CONTINUE
1143) EA(KONT)=.5000*TK1
1144) EB(KONT)=-.5000*TK1-.2500*R4
1145) C(KONT)=0.000
1146) D(KONT)=-.5000*TK2*RAY(I,(J-1),1)-.5000*H(INDEX)*DS*TMPR(I)+
& (.5000*TK2+.5000*H(INDEX)*DS-.5000*FLXB(J)*DS-.2500*R3)
& *RAY(I,J,1)
1147) GO TO 165
1148) 227 CONTINUE
1149) C VERT CONST FLUX, HORIZ SEMI-INF
1150) IFLAG(6)=5
1151) IF(JFLG.NE.1) GO TO 2270
1152) EA(KONT)=TK2*DI2
1153) EB(KONT)=-TK2*DI2-.2500*DI2*R4
1154) C(KONT)=0.00
1155) D(KONT)=-((TK1/(2.00*DI))*RAY((I-1),J,1)-(TK(I,J)/(2.00*DI))*
& TMPB(J)-FLXR(I)*DI2*DS+(TK1/(2.00*DI)+TK(I,J)/(2.00*DI)-
1156) & .2500*DI2*R3)*RAY(I,J,1)
1157) GO TO 165
1158) 2270 CONTINUE
1159) EA(KONT)=TK1/(2.00*DI)
1160) EB(KONT)=-TK1/(2.00*DI)-TK(I,J)/(2.00*DI)-.2500*DI2*R4
1161) C(KONT)=0.00
1162) D(KONT)=-TK2*DI2*RAY(I,(J-1),1)-(TK(I,J)/(2.00*DI))*TMPB(J)-
& FLXR(I)*DI2*DS+(TK2*DI2-.2500*DI2*R3)*RAY(I,J,1)
1163) GO TO 165
1164) 228 CONTINUE
1165) C VERT=SEMI-INF,HORIZ=CONST FLUX
1166) IFLAG(6)=9
1167) IF(JFLG.NE.1) GO TO 2280
1168) EA(KONT)=TK2/(2.000*DI)
1169) EB(KONT)=-((TK2+TK(I,J))/(2.000*DI)-.5000*DI2*R4
1170) C(KONT)=0.000
1171) D(KONT)=-FLXB(J)*DI2*DS-(TK(I,J)/(2.000*DI))*TMPR(I)
& -TK1*DI2*RAY((I-1),J,1)+(TK1*DI2-.5000*DI2*R3)*RAY(I,J,1)
1172) GO TO 165
1173) 2280 CONTINUE
1174) EA(KONT)=TK1*DI2
1175) EB(KONT)=-TK1*DI2-.5000*DI2*R4
1176) C(KONT)=0.000
1177) D(KONT)=-((TK2/(2.000*DI))*RAY(I,(J-1),1)-(TK(I,J)/(2.000*DI))*
& TMPR(I)-FLXB(J)*DI2*DS+((TK2+TK(I,J))/(2.000*DI)-.5000*DI2*R3)
1178) & RAY(I,J,1)
1179) GO TO 165
1180) 229 CONTINUE
1181) C VERT=CONVECT,HORIZ=SEMI-INF
1182) IFLAG(6)=6
1183) IF(JFLG.NE.1) GO TO 2290
1184) EA(KONT)=TK2*DI2
1185) EB(KONT)=-TK2*DI2-H(INDEX)*DI2-.5000*DI2*R4
1186) C(KONT)=0.000
1187) D(KONT)=-H(INDEX)*DI2*TMPR(I)-(TK1/(2.000*DI))*RAY((I-1),J,1)
& -(TK(I,J)/(2.000*DI))*TMPB(J)+((TK1+TK(I,J))/(2.000*DI)
1188) & -.5000*DI2*R3)*RAY(I,J,1)
1189) GO TO 165
1190) 1200)

```

```

1201) 2290 CONTINUE
1202) EA(KONT)=TK1/(2.000*DI)
1203) EB(KONT)=- (TK1+TK(I,J))/(2.000*DI)-.5000*DI2*R4
1204) C(KONT)=0.000
1205) D(KONT)=-H(INDEX)*DI2*TMPR(I)-(TK(I,J)/(2.000*DI))*TMPB(J)
1206) & -TK2*DI2*RAY(I,(J-1),1)+(TK4*DI2+H(INDEX)*DI2-.5000*DI2*R3)*
1207) & RAY(I,J,1)
1208) GO TO 165
1209) 230 CONTINUE
1210) C VERT=SEMI-INF,HORIZ=CONVECT
1211) IFLAG(6)=8
1212) IF(JFLG.NE.1) GO TO 2300
1213) EA(KONT)=TK2/(2.000*DI)
1214) EB(KONT)=- (TK2+TK(I,J))/(2.000*DI)-.5000*DI2*R4
1215) C(KONT)=0.000
1216) D(KONT)=-H(INDEX)*DI2*TMPB(J)-(TK(I,J)/(2.000*DI))*TMPR(I)
1217) & -TK1*DI2*RAY((I-1),J,1)+(H(INDEX)*DI2+TK1*DI2-.5000*DI2*R3)*
1218) & RAY(I,J,1)
1219) GO TO 165
1220) 2300 CONTINUE
1221) EA(KONT)=TK1*DI2
1222) EB(KONT)=-TK1*DI2-.5000*DI2*R4-H(INDEX)*DI2
1223) C(KONT)=0.000
1224) D(KONT)=-H(INDEX)*DI2*TMPB(J)-(TK2/(2.000*DI))*RAY(I,(J-1),1)
1225) & -(TK(I,J)/(2.000*DI))*TMPR(I)+(TK2+TK(I,J))/(2.000*DI)
1226) & -.5000*DI2*R3)*RAY(I,J,1)
1227) GO TO 165
1228) 221 CONTINUE
1229) C CONSTANT CORNER NODE
1230) EB(KONT)=1
1231) D(KONT)=RAY(I,J,1)
1232) IFLAG(6)=10
1233) GO TO 165
1234) 104 CONTINUE
1235) C TOP RIGHT-HAND CORNER
1236) GO TO (231,232,233,234,235,236,237,238,239,240),KRRR(4)
1237) 232 CONTINUE
1238) C CONST FLUX ON BOTH SIDES
1239) IF(JFLG.NE.1) GO TO 2320
1240) EA(KONT)=.500*TK2
1241) EB(KONT)=-.500*TK2-.2500*R4
1242) C(KONT)=0.00
1243) D(KONT)=-.500*TK3*RAY((I+1),J,1)+(.500*TK3-.2500*R3)*
1244) & RAY(I,J,1)-.500*DS*(FLXT(J)+FLXR(I))
1245) GO TO 165
1246) 2320 CONTINUE
1247) EA(KONT)=0.00
1248) EB(KONT)=-.500*TK3-.2500*R4
1249) C(KONT)=.500*TK3
1250) D(KONT)=-.500*TK2*RAY(I,(J-1),1)+(.500*TK2-.2500*R3)*RAY(I,J,1)-
1251) & .500*DS*(FLXT(J)+FLXR(I))
1252) IFLAG(8)=1
1253) GO TO 165
1254) 233 CONTINUE
1255) C CONVECTIVE ON BOTH SIDES
1256) IFLAG(8)=2
1257) IF(JFLG.NE.1) GO TO 2330
1258) EA(KONT)=.500*TK2
1259) EB(KONT)=-.500*TK2-.2500*R4
1260) C(KONT)=0.00
1261) D(KONT)=-.500*TK3*RAY((I+1),J,1)+(.500*TK3+H(INDEX)*DS-.2500*R3)*
1262) & RAY(I,J,1)-.500*H(INDEX)*TMPT(J)-.500*H(INDEX)*TMPR(I)
1263) GO TO 165
1264) 2330 CONTINUE
1265) EA(KONT)=0.00
1266) EB(KONT)=-.500*TK3-.2500*R4
1267) C(KONT)=.500*TK3
1268) D(KONT)=-.500*TK2*RAY(I,(J-1),1)+(.500*TK2+H(INDEX)*DS-.2500*R3)*
1269) & RAY(I,J,1)-.500*H(INDEX)*TMPT(J)-.500*H(INDEX)*TMPR(I)
1270) GO TO 165
1271) 234 CONTINUE
1272) C SEMI-INF ON BOTH SIDES
1273) IFLAG(8)=3
1274) WRITE(1,9997) I,J
1275) GO TO 9999
1276) 236 CONTINUE
1277) C VERT CONST FLUX--HORIZ CONVECT
1278) IFLAG(8)=4
1279) IF(JFLG.NE.1) GO TO 2360
1280) EA(KONT)=.5000*TK2
1281) EB(KONT)=-.5000*TK2-.25000*R4
1282) C(KONT)=0.000
1283) D(KONT)=-.5000*TK3*RAY((I+1),J,1)-.5000*FLXR(I)*DS
1284) & -.5000*H(INDEX)*DS*TMPT(J)+(.5000*TK3+.5000*H(INDEX)*DS
1285) & -.25000*R3)*RAY(I,J,1)
1286) GO TO 165
1287) 2360 CONTINUE
1288) EA(KONT)=0.000
1289) EB(KONT)=-.5000*TK3-.5000*H(INDEX)*DS
1290) C(KONT)=.5000*TK3
1291) D(KONT)=-.5000*TK2*RAY(I,(J-1),1)-.5000*FLXR(I)*DS
1292) & -.5000*H(INDEX)*DS*TMPT(J)+(.5000*TK2-.25000*R3)*RAY(I,J,1)
1293) GO TO 165
1294) C VERT SIDE=SEMI-INF, HORIZ CONST FLUX

```



```

1295) 238 CONTINUE
1296) IF(JFLG.NE.1) GO TO 2380
1297) EA(KONT)=TK2/(2.000*DI)
1298) EB(KONT)=-.5000*(TK(I,J)/DI+TK2/DI+DI2*R4)
1299) C(KONT)=0.000
1300) D(KONT)=-TK3*DI2*RAY((I+1),J,1)-FLXT(J)*DI2*DS+
1301) & (TK3*DI2-.5000*DI2*R3)*RAY(I,J,1)
1302) GO TO 165
1303) 2380 CONTINUE
1304) EA(KONT)=0.000
1305) EB(KONT)=-TK3*DI2-.5000*DI2*R4
1306) C(KONT)=TK3*DI2
1307) D(KONT)=(-TK(I,J)/(2.000*DI))*TMPR(I)-(TK2/(2.000*DI))*
1308) & RAY(I,(J+1),1)+(TK(I,J)/(2.000*DI)+TK2/(2.000*DI)-.5000*DI2
1309) & *R3)*RAY(I,J,1)-FLXT(J)*DI2*DS
1310) GO TO 165
1311) 235 CONTINUE
1312) C VERT CONVECT--HORIZ CONST FLUX
1313) IFLAG(8)=7
1314) IF(JFLG.NE.1) GO TO 2350
1315) EA(KONT)=-.5000*TK2
1316) EB(KONT)=-.5000*TK2-.5000*H(INDEX)*DS-.25000*R4
1317) C(KONT)=0.000
1318) D(KONT)=-.5000*FLXT(J)*DS-.5000*TK3*RAY((I+1),J,1)+
1319) & (.5000*TK3-.25000*R3)*RAY(I,J,1)-.5000*H(INDEX)*DS*TMPR(I)
1320) GO TO 165
1321) 2350 CONTINUE
1322) EA(KONT)=0.000
1323) EB(KONT)=-.5000*TK3-.25000*R4
1324) C(KONT)=-.5000*TK3
1325) D(KONT)=-.5000*TK2*RAY(I,(J-1),1)-.5000*H(INDEX)*DS*
1326) & TMPR(I)+(.5000*TK2+.5000*H(INDEX)*DS-.5000*FLXT(J)*DS
1327) & -.25000*R3)*RAY(I,J,1)
1328) GO TO 165
1329) 240 CONTINUE
1330) C VERT=SEMI-INF,HORIZ=CONVECT
1331) IFLAG(8)=8
1332) IF(JFLG.NE.1) GO TO 2400
1333) EA(KONT)=TK2/(2.000*DI)
1334) EB(KONT)=-TK2+TK(I,J)/(2.000*DI)-.5000*DI2*R4
1335) C(KONT)=0.000
1336) D(KONT)=-H(INDEX)*DI2*TMPT(J)-(TK(I,J)/(2.000*DI))*TMPR(I)
1337) & -TK3*DI2*RAY((I+1),J,1)+(H(INDEX)*DI2+TK3*DI2-.5000*DI2*R3)*
1338) & RAY(I,J,1)
1339) GO TO 165
1340) 2400 CONTINUE
1341) EA(KONT)=0.000
1342) EB(KONT)=-TK3*DI2-.5000*DI2*R4-H(INDEX)*DI2
1343) C(KONT)=TK3*DI2
1344) D(KONT)=-H(INDEX)*DI2*TMPT(J)-(TK2/(2.000*DI))*RAY(I,(J-1),1)
1345) & -(TK(I,J)/(2.000*DI))*TMPR(I)+((TK2+TK(I,J))/(2.000*DI)-.5000*
1346) & DI2*R3)*RAY(I,J,1)
1347) GO TO 165
1348) 237 CONTINUE
1349) 239 CONTINUE
1350) WRITE(1,9997) I,J
1351) GO TO 9999
1352) 231 CONTINUE
1353) C CONSTANT CORNER NODE
1354) EB(KONT)=1
1355) D(KONT)=RAY(I,J,1)
1356) IFLAG(8)=10
1357) GO TO 165
1358) C (((((((((((((((((((((((((((((((((((
1359)
1360)
1361) 165 CONTINUE
1362) C CHECK WHICH PASS
1363) IF(JFLG.EQ.2) GO TO 1602
1364) 170 CONTINUE
1365) C A ROW SET UP ON 1ST PASS NOW.
1366) L=X
1367) IF=1
1368) CALL TRIDIG(L,EA,EB,C,D)
1369) DO 1607 K=1,X
1370) TEMP(I,K)=D(K)
1371) 1607 CONTINUE
1372) 160 CONTINUE
1373) C THE WHOLE GRID 1ST PASS COMPUTED NOW.
1374) JFLG=2
1375) KONT=0
1376) DO 173 I=1,Y
1377) DO 174 J=1,X
1378) OLDT(I,J)=RAY(I,J,1)
1379) RAY(I,J,1)=TEMP(I,J)
1380) 174 CONTINUE
1381) 173 CONTINUE
1382) GO TO 2001
1383) 1602 CONTINUE
1384) 1700 CONTINUE
1385) C A COLUMN SET UP ON 2ND PASS NOW.
1386) 1613 CONTINUE
1387) IF=1
1388) L=Y

```

```

1389) CALL TRIDIG(L,EA,EB,C,D)
1390) DO 1608 K=1,Y
1391) TEMP(K,J)=D(K)
1392) 1608 CONTINUE
1393) 1600 CONTINUE
1394) C WHOLE GRID 2ND PASS COMPLETE NOW.
1395) DO 1609 J=1,X
1396) DO 1610 I=1,Y
1397) RAY(I,J,1)=TEMP(I,J)
1398) C CHECK IF FRONT SKIPPED THE NODE
1399) K=RAY(I,J,3)
1400) CC=TPC(K)+(TDEL/2.D0)
1401) DD=TPC(K)-(TDEL/2.D0)
1402) IF((QLDT(I,J).GT.CC).AND.(TEMP(I,J).LT.DD)) GO TO 188
1403) IF((QLDT(I,J).LT.DD).AND.(TEMP(I,J).GT.CC)) GO TO 189
1404) GO TO 1610
1405) 188 CONTINUE
1406) C SKIPPED FROM UNFROZEN TO FROZEN
1407) RAY(I,J,1)=CC+(CP(I,J)/CPPC(K))*(RAY(I,J,1)-CC)
1408) GO TO 1610
1409) 189 CONTINUE
1410) CC SKIPPED FROM FROZEN TO UNFROZEN
1411) RAY(I,J,1)=DD+(CP(I,J)/CPPC(K))*(RAY(I,J,1)-DD)
1412) 1610 CONTINUE
1413) 1609 CONTINUE
1414) JFLG=1
1415) KONT=0
1416) Q=Q+1
1417) ITCC=ITCC+1
1418) ITPP=ITPP+1
1419) C
1420) C *****
1421) C WRITE OUTPUT
1422) C ITRT IS NO. OF TIME STEPS BEFORE TEMPERATURES PRINTED
1423) C ITCC COUNTS FROM 1 TO ITRT
1424) C ITPC IS NO. OF TIME STEPS BEFORE ISOTHM IS CALLED
1425) C ITPP COUNTS FROM 1 TO ITPC
1426) C
1427) C TT=Q*DELT
1428) C
1429) C FIGURE & WRITE ISOTHERM LOCATIONS
1430) IF(ITPP.NE.ITPC) GO TO 920
1431) CALL ISOTHM
1432) ITPP=0
1433) 920 CONTINUE
1434) C
1435) C WRITE TEMPERATURE DISTRIBUTIONS
1436) IF(ITCC.NE.ITRT) GO TO 888
1437) WRITE(11,893) Q,TT
1438) 893 FORMAT(/,/,,1X,'TEMPERATURES AFTER',I4,' TIME STEPS (',
1439) & F12.3,' HRS):')
1440) RRX=X/17
1441) JJX=RRX
1442) JJR=X-(17*JJX)
1443) DO 531 JJ=1,JJX
1444) IF(JJX.EQ.0) GO TO 531
1445) J2=17*JJ
1446) J1=J2-16
1447) WRITE(11,500) ((RAY(I,J,1),J=J1,J2),I=1,Y)
1448) IF(JJ.LT.JJX) WRITE(11,503)
1449) 531 CONTINUE
1450) J2=X
1451) J1=X-JJR+1
1452) IF(JJR.EQ.0) GO TO 532
1453) IF(JJX.NE.0) WRITE(11,503)
1454) DO 532 I=1,Y
1455) WRITE(11,500) (RAY(I,J,1),J=J1,J2)
1456) 532 CONTINUE
1457) ITCC=0
1458) 888 CONTINUE
1459) C *****
1460) C
1461) C CHECK NO. OF WHOLE TIME STEPS AND GO BACK TO START IF NOT DONE
1462) IF(Q.GE.IMAX) GO TO 172
1463) JFLG=1
1464) KONT=0
1465) GO TO 2002
1466) 172 CONTINUE
1467) 200 CONTINUE
1468) C WRITE BOUNDARY TYPES INTO RESULT
1469) WRITE(6,404)
1470) 404 FORMAT(/,2X,'BOUNDARY CONDITIONS:')
1471) DO 410 K=1,8
1472) KKK=K
1473) IF(IFLAG(K).EQ.0) GO TO 410
1474) GO TO (400,450,401,451,402,452,403,453),KKK
1475) 400 WRITE(6,405)
1476) 405 FORMAT(/,1X,'TOP BOUNDARY')
1477) GO TO 470
1478) 401 WRITE(6,406)
1479) 406 FORMAT(1X,'LEFT BOUNDARY')
1480) GO TO 470
1481) 402 WRITE(6,407)
1482) 407 FORMAT(1X,'BOTTOM BOUNDARY')

```

```

1483) GO TO 470
1484) 403 WRITE(6,408)
1485) 408 FORMAT(1X,'RIGHT BOUNDARY')
1486) GO TO 470
1487) 450 WRITE(6,460)
1488) 460 FORMAT(1X,'TOP LEFT CORNER')
1489) GO TO 470
1490) 451 WRITE(6,461)
1491) 461 FORMAT(1X,'BOTTOM LEFT CORNER')
1492) GO TO 470
1493) 452 WRITE(6,462)
1494) 462 FORMAT(1X,'BOTTOM RIGHT CORNER')
1495) GO TO 470
1496) 453 WRITE(6,463)
1497) 463 FORMAT(1X,'TOP RIGHT CORNER')
1498) GO TO 470
1499) 470 CONTINUE
1500) GO TO (471,472,473,474,475,476,477,478,479,480,481,482,
1501) & 483),IFLAG(K)
1502) 471 WRITE(6,492)
1503) 492 FORMAT(1H+,'
CONST FLUX ON BOTH SIDES')
1504) GO TO 410
1505) 472 WRITE(6,493)
1506) 493 FORMAT(1H+,'
CONVECTIVE ON BOTH SIDES')
1507) GO TO 410
1508) 473 WRITE(6,494)
1509) 494 FORMAT(1H+,'
SEMI-INF ON BOTH SIDES')
1510) GO TO 410
1511) 474 WRITE(6,484)
1512) 484 FORMAT(1H+,'
VERT CONST FLUX--HORIZ CONVECT')
1513) GO TO 410
1514) 475 WRITE(6,485)
1515) 485 FORMAT(1H+,'
VERT CONST FLUX--HORIZ SEMI-INF')
1516) GO TO 410
1517) 476 WRITE(6,486)
1518) 486 FORMAT(1H+,'
VERT CONVECT--HORIZ SEMI-INF')
1519) GO TO 410
1520) 477 WRITE(6,487)
1521) 487 FORMAT(1H+,'
VERT CONVECT--HORIZ CONST FLUX')
1522) GO TO 410
1523) 478 WRITE(6,488)
1524) 488 FORMAT(1H+,'
VERT SEMI-INF--HORIZ CONVECT')
1525) GO TO 410
1526) 479 WRITE(6,489)
1527) 489 FORMAT(1H+,'
VERT SEMI-INF--HORIZ CONST FLUX')
1528) GO TO 410
1529) 480 WRITE(6,491)
1530) 491 FORMAT(1H+,'
CONSTANT TEMPERATURE')
1531) GO TO 410
1532) 481 CONTINUE
1533) WRITE(6,416)
1534) GO TO 410
1535) 482 CONTINUE
1536) WRITE(6,417)
1537) GO TO 410
1538) 483 CONTINUE
1539) WRITE(6,418)
1540) GO TO 410
1541) 416 FORMAT(1H+,'
SEMI-INFINITE')
1542) 417 FORMAT(1H+,'
CONVECTIVE SURFACE')
1543) 418 FORMAT(1H+,'
CONSTANT HEAT FLUX')
1544) 410 CONTINUE
1545) WRITE(1,183) Q
1546) WRITE(6,183) Q
1547) 183 FORMAT(/,1X,'NUMBER OF TIME STEPS=',I5)
1548) DD=Q*DELT
1549) RRX=X/17
1550) JJX=RRX
1551) JJR=X-(17*JJX)
1552) C
1553) C WRITE THE FINAL TEMPS & THERMAL VALUES
1554) DO 510 KK=1,5
1555) GO TO (511,512,513,514,515),KK
1556) 511 WRITE(6,893) Q,DD
1557) GO TO 516
1558) 512 WRITE(6,911)
1559) GO TO 516
1560) 513 WRITE(6,912)
1561) GO TO 516
1562) 514 WRITE(6,914)
1563) GO TO 516
1564) 515 WRITE(6,913)
1565) GO TO 516
1566) 516 DO 518 JJ=1,JJX
1567) IF(JJX.EQ.0) GO TO 518
1568) J2=17*JJ
1569) J1=J2-16
1570) GO TO (519,520,521,522,523),KK
1571) 519 WRITE(6,500) ((RAY(I,J,1),J=J1,J2),I=1,Y)
1572) GO TO 517
1573) 520 WRITE(6,500) ((TK(I,J),J=J1,J2),I=1,Y)
1574) GO TO 517
1575) 521 WRITE(6,500) ((CP(I,J),J=J1,J2),I=1,Y)
1576) GO TO 517

```

```

1577) 522 WRITE(6,500) ((RO(I,J),J=J1,J2),I=1,Y)
1578) GO TO 517
1579) 523 WRITE(6,502) ((ISTAT(I,J),J=J1,J2),I=1,Y)
1580) 517 IF(JJ.LT.JJX) WRITE(6,503)
1581) 518 CONTINUE
1582) J2=X
1583) J1=X-JJR+1
1584) IF(JJR.EQ.0) GO TO 524
1585) IF(JJX.NE.0) WRITE(6,503)
1586) DO 524 I=1,Y
1587) GO TO (525,526,527,528,529),KK
1588) 525 WRITE(6,500) (RAY(I,J,1),J=J1,J2)
1589) GO TO 524
1590) 526 WRITE(6,500) (TK(I,J),J=J1,J2)
1591) GO TO 524
1592) 527 WRITE(6,500) (CP(I,J),J=J1,J2)
1593) GO TO 524
1594) 528 WRITE(6,500) (RO(I,J),J=J1,J2)
1595) GO TO 524
1596) 529 WRITE(6,502) (ISTAT(I,J),J=J1,J2)
1597) 524 CONTINUE
1598) 510 CONTINUE
1599) 500 FORMAT(1X,17F7.2)
1600) 911 FORMAT(/,1X,'FINAL TK(I,J):')
1601) 912 FORMAT(/,1X,'FINAL CP(I,J):')
1602) 501 FORMAT(1X,17F7.2)
1603) 914 FORMAT(/,1X,'FINAL RO(I,J):')
1604) 913 FORMAT(/,1X,'FINAL ISTAT(I,J):')
1605) 502 FORMAT(1X,17I3)
1606) 503 FORMAT(/,3X,'AND THE NEXT SET OF COLUMNS:')
1607) WRITE(11,915) CPPC(1),ROPC(1)
1608) WRITE(11,916) TPC(1),TDEL
1609) 915 FORMAT(/,1X,'CPPC(1)=',F6.2,' ROPC(1)=',F6.2)
1610) 916 FORMAT(1X,'TPC(1)=',F6.2,' TDEL=',F6.2)
1611) 9998 CONTINUE
1612) 927 FORMAT(/,/)
1613) C CREATE A NEW DATA FILE WITH END RESULTS
1614) WRITE(9,10) DS,DELT,DI,TDEL
1615) WRITE(9,20) A,X,Y,NISO,ITRT,IMAX,ITPC
1616) WRITE(9,16) ((ROPC(K),CPPC(K),HL(K),TPC(K)),K=1,A)
1617) WRITE(9,15) (TISO(B),B=1,NISO)
1618) WRITE(9,30) (H(L),L=1,A)
1619) WRITE(9,35) ((RAY(I,J,3),J=1,X),I=1,Y)
1620) WRITE(9,35) ((RAY(I,J,2),J=1,X),I=1,Y)
1621) WRITE(9,35) ((RAY(I,J,1),J=1,X),I=1,Y)
1622) WRITE(9,34) (KRNR(J),J=1,4)
1623) WRITE(9,35) (FLXT(J),J=1,X)
1624) WRITE(9,35) (FLXB(J),J=1,X)
1625) WRITE(9,35) (FLXL(I),I=1,Y)
1626) WRITE(9,35) (FLXR(I),I=1,Y)
1627) WRITE(9,35) (TMPT(J),J=1,X)
1628) WRITE(9,35) (TMPB(J),J=1,X)
1629) WRITE(9,35) (TMPL(I),I=1,Y)
1630) WRITE(9,35) (TPMR(I),I=1,Y)
1631) 9997 FORMAT(1X,'NO EQUATION FOR RAY(',I2,',',I2,',').')
1632) 9999 CONTINUE
1633) C CLOSE FILES
1634) CALL CONTRL(4,'ADPTMP',11)
1635) CALL CONTRL(4,'ADPDAT',5)
1636) CALL CONTRL(4,'ADPOUT',6)
1637) CALL CONTRL(4,'ADPNOT',9)
1638) CALL CONTRL(4,'POINT1',13)
1639) CALL EXIT
1640) END
1641) C
1642) C *****
1643) SUBROUTINE TRIDIG(N,A,B,C,D)
1644) IMPLICIT INTEGER*2(I,N)
1645) IMPLICIT DOUBLE PRECISION(A,B,C,D)
1646) DIMENSION A(1),B(1),C(1),D(1)
1647) C ADJUST COEFFS FOR B&D FROM ELIMINATING A
1648) DO 10 I=2,N
1649) AB=A(I)/B(I-1)
1650) B(I)=B(I)-AB*C(I-1)
1651) D(I)=D(I)-AB*D(I-1)
1652) 10 CONTINUE
1653) C BACK SUBSTITUTE
1654) N1=N-1
1655) D(N)=D(N)/B(N)
1656) DO 20 I=1,N1
1657) M=N-I
1658) D(M)=(D(M)-C(M)*D(M+1))/B(M)
1659) 20 CONTINUE
1660) RETURN
1661) END
1662) C
1663) C *****
1664) SUBROUTINE ISOTHM
1665) C THIS PROGRAM FINDS ISOTHERMS IN RAY(I,J,1).
1666) IMPLICIT INTEGER*2(A,B,I-L,N,G,W,X,Y,Z)
1667) IMPLICIT DOUBLE PRECISION(C-H,M,O,P,R-V)
1668) DIMENSION EXRY(9,200),EYRY(9,200),COUNT(9)
1669) COMMON/M12I/ X,Y,NISO,Q

```

```

1670) COMMON/M12R/ RAY(80,80,3),TISO(9),DS,DELTA
1671) C NISO=NUMBER OF ISOTHERMS
1672) C COUNT(B)=COUNTER FOR NO. ELTS IN EACH ISOTHERM
1673) C TISO(B)=TEMP OF EACH ISOTHERM
1674) C LET TISO(1) BE THE HOTTEST ISOTHERM
1675) C EXRY(B,K)= ARRAY FOR X-COORDINATES
1676) C B INDICATES WHICH ISOTHERM
1677) C K=COUNT(B) & INDICATES POSITION OF PT IN ISOTHERM LIST
1678) C
1679) C
1680) C CHANGE THE NEXT STATEMENT TO AGREE WITH DIMENSION
1681) DO 6 K=1,200
1682) DO 7 B=1,NISO
1683) EXRY(B,K)=0
1684) EYRY(B,K)=0
1685) 7 CONTINUE
1686) 6 CONTINUE
1687) DO 8 K=1,9
1688) COUNT(K)=0.
1689) 8 CONTINUE
1690) CNTPT=0.
1691) C LOOP TO LOCATE ISOTHERMS
1692) C FOR LEFT SEMI-INNF,LET XL=3
1693) C FOR BOTTOM SEMI-INF,LET YB=Y-3
1694) C FOR RIGHT SEMI-INF,LET XR=X-3, AND COMMENT OUT LOOP 800
1695) C FOR TOP SEMI-INF,LET YT=3, AND COMMENT OUT LOOP 860
1696) C IF NOT USING SEMI-INF,XL=1,XR=X-1,YT=2,YB=Y
1697) XL=1
1698) XR=X-1
1699) YB=Y-3
1700) YT=2
1701) DO 100 I=YT,YB
1702) DO 200 J=XL,XR
1703) C
1704) C EXAMINE TEMPS HORIZONTALLY
1705) RJ=RAY(I,J,1)
1706) RJ1=RAY(I,(J+1),1)
1707) IF((RJ.GT.TISO(1)).AND.(RJ1.GT.TISO(1))) GO TO 500
1708) IF((RJ.LT.TISO(NISO)).AND.(RJ1.LT.TISO(NISO))) GO TO 500
1709) DO 500 B=1,NISO
1710) IF((TISO(B).GT.RJ).AND.(TISO(B).LT.RJ1)) GO TO 400
1711) IF((TISO(B).LT.RJ).AND.(TISO(B).GT.RJ1)) GO TO 400
1712) IF(TISO(B).EQ.RJ) GO TO 402
1713) GO TO 500
1714) 400 CONTINUE
1715) 401 FORMAT(1X,'RAY(*,I3,I3,*,1)=*,F6.2)
1716) EXCD=(RAY(I,J,1)-TISO(B))/(RAY(I,J,1)-RAY(I,(J+1),1))+J
1717) EXCO=DS*(EXCD-1.)
1718) COUNT(B)=COUNT(B)+1
1719) K=COUNT(B)
1720) EXRY(B,K)=EXCO
1721) EYRY(B,K)=DS*(I-1.)
1722) CNTPT=CNTPT+1
1723) GO TO 500
1724) 402 CONTINUE
1725) COUNT(B)=COUNT(B)+1
1726) K=COUNT(B)
1727) EXRY(B,K)=DS*(J-1.)
1728) EYRY(B,K)=DS*(I-1.)
1729) CNTPT=CNTPT+1
1730) 500 CONTINUE
1731) C EXAMINE TEMPS VERTICALLY
1732) RI=RAY(I,J,1)
1733) RI1=RAY(I-1,J,1)
1734) IF((RI.GT.TISO(1)).AND.(RI1.GT.TISO(1))) GO TO 525
1735) IF((RI.LT.TISO(NISO)).AND.(RI1.LT.TISO(NISO))) GO TO 525
1736) DO 525 B=1,NISO
1737) IF((TISO(B).GT.RI).AND.(TISO(B).LT.RI1)) GO TO 550
1738) IF((TISO(B).LT.RI).AND.(TISO(B).GT.RI1)) GO TO 550
1739) GO TO 525
1740) 550 CONTINUE
1741) 404 FORMAT(1X,'RJ=*,F6.2,* RJ1=*,F6.2)
1742) EYCD=(RAY((I-1),J,1)-TISO(B))/(RAY((I-1),J,1)-
1743) & RAY(I,J,1))+I-1.)
1744) EYCO=DS*(EYCD-1.)
1745) COUNT(B)=COUNT(B)+1
1746) K=COUNT(B)
1747) EYRY(B,K)=EYCO
1748) EXRY(B,K)=DS*(J-1.)
1749) CNTPT=CNTPT+1
1750) 525 CONTINUE
1751) 200 CONTINUE
1752) 100 CONTINUE
1753) C 800 LOOP IS FOR RIGHT HAND SIDE
1754) DO 800 I=YT,YB
1755) DO 851 B=1,NISO
1756) RI=RAY(I,X,1)
1757) RI1=RAY(I-1,X,1)
1758) IF((RI.GT.(TISO(B))).AND.(RI1.LT.(TISO(B)))) GO TO 852
1759) IF((RI.LT.(TISO(B))).AND.(RI1.GT.(TISO(B)))) GO TO 852
1760) IF(RI.EQ.TISO(B)) GO TO 853

```

```

1761) GO TO 851
1762) 852 CONTINUE
1763) EYCD=(I-1.)+(RI1-TISO(B))/(RI1-RAY(I,X,1))
1764) EYCO=DS*(EYCD-1.)
1765) COUNT(B)=COUNT(B)+1
1766) K=COUNT(B)
1767) EYRY(B,K)=EYCO
1768) EXRY(B,K)=DS*(X-1.)
1769) CNTPT=CNTPT+1
1770) GO TO 851
1771) 853 CONTINUE
1772) COUNT(B)=COUNT(B)+1
1773) K=COUNT(B)
1774) EYRY(B,K)=DS*(I-1.)
1775) EXRY(B,K)=DS*(X-1.)
1776) CNTPT=CNTPT+1
1777) 851 CONTINUE
1778) 800 CONTINUE
1779) C 860 LOOP IS FOR TOP ROW
1780) XX=X-1
1781) I=1
1782) DO 860 J=XL,XR
1783) RJ=RAY(1,J,1)
1784) RJ1=RAY(1,(J+1),1)
1785) DO 861 B=1,NISO
1786) IF((TISO(B).GT.RJ).AND.(TISO(B).LT.RJ1)) GO TO 862
1787) IF((TISO(B).LT.RJ).AND.(TISO(B).GT.RJ1)) GO TO 862
1788) IF(RJ.EQ.TISO(B)) GO TO 863
1789) GO TO 861
1790) 862 CONTINUE
1791) EXCD=(RAY(I,J,1)-TISO(B))/(RAY(I,J,1)-RAY(I,(J+1),1))+J
1792) EXCO=DS*(EXCD-1.)
1793) COUNT(B)=COUNT(B)+1
1794) K=COUNT(B)
1795) EXRY(B,K)=EXCO
1796) EYRY(B,K)=0
1797) CNTPT=CNTPT+1
1798) GO TO 861
1799) 863 CONTINUE
1800) COUNT(B)=COUNT(B)+1
1801) K=COUNT(B)
1802) EXRY(B,K)=DS*(J-1.)
1803) EYRY(B,K)=0
1804) CNTPT=CNTPT+1
1805) 861 CONTINUE
1806) 860 CONTINUE
1807) C
1808) C
1809) TIM=Q*DELT
1810) WRITE(13,96) CNTPT,TIM
1811) 96 FORMAT(1X,'THE FOLLOWING',I5,' POINTS REPRESENT TIME=',
1812) & F8.4,' HOURS:')
1813) 94 FORMAT(1X,F7.2)
1814) 92 FORMAT(1X,I5)
1815) DO 666 B=1,NISO
1816) L=COUNT(B)
1817) IF(L.EQ.0) GO TO 99
1818) WRITE(13,91) L,B,TISO(B)
1819) 91 FORMAT(1X,2I6,F6.2)
1820) 99 CONTINUE
1821) DO 95 K=1,L
1822) IF(L.EQ.0) GO TO 95
1823) WRITE(13,90) (EXRY(B,K),EYRY(B,K))
1824) 95 CONTINUE
1825) 90 FORMAT(1X,2F15.6)
1826) 666 CONTINUE
1827) 93 FORMAT(1X,'TOTAL NO. POINTS FOUND=',I5)
1828) RETURN
1829) END
1830) C *****
1831) C *****
1832) C ADDATA, DATA-GATHERING SUBROUTINE
1833) SUBROUTINE ADDATA
1834) IMPLICIT INTEGER*2(A,B,I-L,N,Q,W,X,Y,Z)
1835) IMPLICIT DOUBLE PRECISION(C-H,M,O,P,R-V)
1836) COMMON/M12I/ X,Y,NISO,Q
1837) COMMON/M12R/ RAY(80,80,3),TISO(9),DS,DELT
1838) COMMON/M1I/ IMAX,A,ITRT,KRNR(4),ITPC
1839) COMMON/M1R/ DI,TDEL,H(2),
1840) & FLXT(80),FLXB(80),FLXL(80),FLXR(80),TMPT(80),
1841) & TMPB(80),TMPL(80),TMPR(80),ROPC(2),CPPC(2),HL(2),TPC(2)
1842) C ADPDAT IS THE DATA FILE FOR ADIPC
1843) C ALL UNITS ARE IN METERS,HOURS,CELSIUS
1844) C DELT=TIME INCREMENT (HRS)
1845) C IMAX=MAX NO. OF TIME STEPS REQUESTED
1846) C DS=DISTANCE BETWEEN NODES (METERS)
1847) C DI=THE MULTIPLE OF DS WHICH IS HALF THE DISTANCE TO INFINIY.
1848) C (DS*DI=HALF THE DISTANCE TO INFINITY) DI>1
1849) C TSRF=SURFACE TEMP (OUTSIDE GRID) (INFINITE DIST AWAY)
1850) C TBTM=BOTTOM TEMP (UNDER GRID) (INFINITE DIST AWAY)
1851) C TRIT=TEMP TO RIGHT OF GRID (INFINITE DIST)
1852) C TLFY=TEMP TO LEFT OF GRID (INFINITE DIST AWAY)
1853) C X= NO. OF GRID NODES HORIZONTALLY (X>=2)
1854) C Y= NO. OF GRID NODES VERTICALLLY (Y>=2)
1855) C A= NO. OF DIFFERENT MATERIALS IN THE GRID

```

```

1856) C CP(I,J)=SPECIFIC HEAT (W*HR/KG*K)
1857) C TK(I,J)=THERMAL CONDUCTIVITY (W/M*K)
1858) C RO(I,J)=DENSITY (KG/M3)
1859) C H(L)=CONVECTION COEFFICIENT
1860) C CPO=SPECIFIC HEAT OF PREVIOUS TIME STEP (FIGURED IN MAIN PROGRAM)
1861) C ROO=DENSITY OF PREVIOUS TIME STEP (FIGURED IN MAIN PROGRAM)
1862) C RAY(I,J,1)=PRESENT NODAL TEMP
1863) C RAY(I,J,2)=LOCATION TYPE:
1864) C THIS PROGRAM HAS SEMI-INFINITE OPTION ON LEFT,
1865) C BOTTOM, AND RIGHT SIDES ONLY, AT PRESENT.
1866) C ASTERISKS DENOTE SITUATIONS NOT YET PROGRAMMED.
1867) C =1 TOP LEFT CORNER
1868) C =2 BOTTOM LEFT CORNER
1869) C =3 BOTTOM RIGHT CORNER
1870) C =4 TOP RIGHT CORNER
1871) C =5 INTERIOR NODE, FLUCTUATING TEMP
1872) C =6 INTERIOR NODE, CONSTANT TEMP
1873) C =7 NODE ON CONST TEMP BOUNDARY
1874) C =8 CONST FLUX BOUNDARY: TOP
1875) C =27 CONST FLUX BOUNDARY: LEFT
1876) C =28 CONST FLUX BOUNDARY: BOTTOM
1877) C =29 CONST FLUX BOUNDARY: RIGHT
1878) C =9 CONVECTIVE BOUNDARY: TOP
1879) C =30 CONVECTIVE BOUNDARY: LEFT
1880) C =31 CONVECTIVE BOUNDARY: BOTTOM
1881) C =32 CONVECTIVE BOUNDARY: RIGHT
1882) C =10 NODE ON SEMI-INFINITE BOUNDARY
1883) C =11 NODE ADJACENT TO LEFT SEMI-INF BNDRY
1884) C =12 NODE ADJACENT TO BOTTOM SEMI-INF BNDARY
1885) C =13 NODE ADJACENT TO RIGHT SEMI-INF BNDRY
1886) C =14 NODE ADJACENT TO TOP SEMI-INF BNDRY *
1887) C THE NEXT FIVE CONDITIONS NEEDED WITH SEMI-INF CORNER
1888) C =15 SQUARE NODE ADJACENT TO TWO SEMI-INF SIDES
1889) C =16 SEMI-INF NODE ABOVE SEMI-INF CORNER NODE
1890) C (FOR BOTTOM LEFT & RIGHT ONLY, AT PRESENT)
1891) C =17 SEMI-INF NODE TO LEFT OF BOTTOM RIGHT SEMI-INF CORNER NODE
1892) C =18 SEMI-INF NODE BELOW SEMI-INF CORNER NODE *
1893) C =19 SEMI-INF NODE TO RIGHT OF BOTTOM LEFT SEMI-INF CORNER NODE
1894) C THE NEXT FOUR CONDITNS NEEDED WITH CORNER W ONE SIDE SEMI-INF,
1895) C ONE SIDE CONST FLUX
1896) C =20 TOP CONST FLUX NODE ADJACENT TO RIGHT SEMI-INF
1897) C =21 RIGHT SIDE CONST FLUX NODE ADJACENT TO BOTTOM SEMI-INF
1898) C =22 TOP CONST FLUX NODE ADJACENT TO LEFT SEMI-INF
1899) C =23 LEFT SIDE CONST FLUX NODE ADJACENT TO BOTTOM SEMI-INF
1900) C THE NEXT THREE CONDITNS NEEDED WITH CORNER W ONE SIDE SEMI-INF,
1901) C ONE SIDE CONVECTIVE
1902) C =24 RIGHT SIDE CONVECT NODE ADJ TO BOTTOM SEMI-INF*
1903) C =25 TOP CONVECT NODE ADJ TO LEFT SEMI-INF*
1904) C =26 LEFT SIDE CONST FLUX NODE ADJACENT TO BOTTOM SEMI-INF*
1905) C RAY(I,J,3)= INDEX OF MATERIAL (RANGES FROM 1 TO A)
1906) C KRNR(1)=TOP LEFT CORNER
1907) C KRNR(2)=BOTTOM LEFT CORNER
1908) C KRNR(3)=BOTTOM RIGHT CORNER
1909) C KRNR(4)=TOP RIGHT CORNER
1910) C KRNR(L)=1 CONSTANT TEMP CORNER NODE
1911) C =2 CONSTANT FLUX (ON BOTH SIDES) CORNER NODE
1912) C =3 CONVECTIVE (ON BOTH SIDES) CORNER NODE
1913) C =4 SEMI-INFINITE (ON BOTH SIDES) CORNER NODE
1914) C =5 VERTICAL SIDE=CONVECT, HORIZ=CONST FLUX
1915) C =6 VERTICAL SIDE=CONST FLUX, HORIZ =CONVECT
1916) C =7 VERTICAL SIDE =CONST FLUX, HORIZ =SEMI-INF
1917) C =8 VERTICAL SIDE=SEMI-INF, HORIZ=CONST FLUX
1918) C (TOP LEFT & TOP RIGHT ONLY AT PRESENT)
1919) C =9 VERTICAL SIDE =CONVECT, HORIZ=SEMI-INF *
1920) C =10 VERTICAL SIDE=SEMI-INF, HORIZ=CONVECT*
1921) C TEMP(I,J)= TEMP AT TIME (T+DELT)
1922) C ISTAT(I,J)=INDEX OF STATE FOR TEMP(I,J)
1923) C =1 FROZEN
1924) C =2 UNFROZEN
1925) C =3 UNDERGOING PHASE CHANGE
1926) C FLXT(J)=FLUX FROM TOP OF GRID
1927) C FLXB(J)=FLUX FROM BOTTOM OF GRID
1928) C FLXL(I)=FLUX FROM LEFT SIDE OF GRID
1929) C FLXR(I)=FLUX FROM RIGHT SIDE OF GRID
1930) C TMPT(J)=TEMP OUTSIDE GRID FOR TOP
1931) C TMPB(J)=TEMP OUTSIDE GRID FOR BOTTOM
1932) C TMPL(I)=TEMP OUTSIDE GRID FOR LEFT
1933) C TMPR(I)=TEMP OUTSIDE GRID FOR RIGHT
1934) C NISO= NO. OF ISOTHERMS TO BE PLOTTED
1935) C TMAX= TEMP OF HOTTEST ISOTHERM
1936) C TMIN= TEMP OF COLDEST ISOTHERM
1937) C TISO(B)= ARRAY CONTAINING ISOTHERM TEMPS
1938) C TISO(1) HAS THE HOTTEST ISOTHERM
1939) C COUNT(B)=COUNTER FOR NO.ELTS IN EACH ISOTHERM
1940) C IIRT=NO. TIME STEPS BEFORE TEMPS PRINTED
1941) C ITPC=NO. TIME STEPS BEFORE ISOTHM IS CALLED
1942) C FOR THE NEXT 4 STMTS,K=RAY(I,J,3)
1943) C ROPC(K)=DENSITY AT PHASE CHANGE FRONT(KG/M3)
1944) C HL(K)=LATENT HEAT OF FUSION (W*HR/KG*K)
1945) C TPC(K)=TEMPERATURE OF PHASE CHANGE (°C)
1946) C CPPC(K)=SPECIFIC HEAT DURING PHASE CHANGE
1947) C TDEL=TOTAL TEMP RANGE FOR TPC
1948) C (EX: IF TPC=0°,TDEL=2,PHSCH OCCURS FROM -1° TO 1°)

```

```

1947) C INITIALIZE VARIABLES AND ARRAYS
1948)   TSRF=-4.67000
1949)   TBTM=4.6700
1950)   TLFT=4.67000
1951)   TRIT=4.67000
1952)   DS=.005000
1953)   DELT=.0025000
1954)   DI=50.000
1955)   A=1
1956)   IMAX=1200
1957)   X=3
1958)   Y=25
1959)   TDEL=1.000
1960)   NISO=1
1961)   ITRT=40
1962)   ITPC=10
1963) C
1964) C DO NOT CHANGE THE FOLLOWING 6 LINES.
1965)   ETIME=ITRT*DELT
1966)   MTIME=IMAX*DELT
1967)   WRITE(1,3) ITRT,ETIME,IMAX,MTIME
1968) 3   FORMAT(1X,'TEMPERATURES WILL BE PRINTED EVERY',I3,' ITNS',/,
1969)   C 1X,' (EVERY',F12.4,' HRS.)',/,1X,
1970)   C 'MAX NO. ITNS=',I4,' (=',F12.4,' HRS)')
1971) C
1972) C SET UP MATERIAL PROPERTIES FOR EACH MATERIAL.
1973) C (YOU SET THE VALUES OF K & CP & RO IN MAIN PROGRAM.)
1974)   H(A)=0.00
1975) C PROPERTIES FOR PHASE CHANGE:
1976)   ROPC(1)=917.000
1977)   HL(1)=93.000
1978)   TPC(1)=0.000
1979) C THE FOLLOWING SPECIFIC HEAT SHOULD INCLUDE LATENT HEAT.
1980) C YOU CALCULATE IT BY: CP=(CP(FROZ)+CP(UNFROZ))/2 + HL(K)/TDEL
1981)   DO 2 K=1,A
1982)   CPPC(K)=093.87000
1983) 2   CONTINUE
1984) C
1985) C SPECIFY THE ISOTHERMS TO BE LOCATED IN SUBROUTINE ISOTHM
1986) C IN ORDER, WITH TISO(1) THE HOTTEST.
1987)   TISO(1)=0.000
1988) C
1989) C INDICATE THE FLUXES FROM THE SIDES OF THE GRID.
1990) C THIS IS USED ONLY FOR THE CONSTANT FLUX BOUND.CONDITION.
1991)   DO 11 J=1,X
1992)   FLXT(J)=0.00
1993)   FLXB(J)=0.00
1994) 11  CONTINUE
1995)   DO 12 I=1,Y
1996)   FLXL(I)=0.00
1997)   FLXR(I)=0.00
1998) 12  CONTINUE
1999) C
2000) C INDICATE THE TEMPERATURES OUTSIDE THE GRID.
2001) C USED FOR SEMI-INF & CONVECT BOUNDARYS
2002)   DO 13 J=1,X
2003)   TMPT(J)=TSRF
2004)   TMPB(J)=TBTM
2005) 13  CONTINUE
2006)   DO 14 I=1,Y
2007)   TMPL(I)=TLFT
2008)   TMPR(I)=TRIT
2009) 14  CONTINUE
2010) C
2011) C SET UP RAY(I,J,3) INDEX OF MATERIALS
2012) C THERE ARE A MATERIALS. LET THE SURROUNDING
2013) C MATERIAL BE THE 'ATH' MATERIAL. START
2014) C WITH '1'.
2015) C THIS LOOP ASSIGNS THE 'ATH' MATERIAL TO
2016) C THE REMAINING NODES.
2017)   DO 10 J=1,X
2018)   DO 20 I=1,Y
2019)   RAY(I,J,3)=A
2020) 20  CONTINUE
2021) 10  CONTINUE
2022) C
2023) C SET UP RAY(I,J,2) NODAL LOCATION TYPE
2024) C TOP BOUNDARY
2025)   DO 40 J=1,X
2026)   RAY(1,J,2)=7
2027) 40  CONTINUE
2028) C BOTTOM BOUNDARY
2029)   DO 50 J=1,X
2030)   RAY(Y,J,2)=10
2031) 50  CONTINUE
2032) C LEFT BOUNDARY
2033)   DO 60 I=1,Y
2034)   RAY(I,1,2)=27
2035) 60  CONTINUE
2036) C RIGHT BOUNDARY
2037)   DO 70 I=1,Y
2038)   RAY(I,X,2)=29
2039) 70  CONTINUE
2040) C SET UP CORNER CONDITIONS

```



```

2041) C  --DON'T CHANGE THE "RAY(I,J,2)" STATEMENTS.
2042)     RAY(1,1,2)=1
2043)     RAY(Y,1,2)=2
2044)     RAY(Y,X,2)=3
2045)     RAY(1,X,2)=4
2046) C      TOP LEFT CORNER
2047)     KRNR(1)=1
2048)     RAY(1,1,1)=TSRF
2049) C      BOTTOM LEFT CORNER
2050)     KRNR(2)=7
2051)     RAY(Y,1,1)=TBTM
2052) C      BOTTOM RIGHT CORNER
2053)     KRNR(3)=7
2054)     RAY(Y,X,1)=TBTM
2055) C      TOP RIGHT CORNER
2056)     KRNR(4)=1
2057)     RAY(1,X,1)=TSRF
2058) C
2059) C  INTERIOR NODES GIVEN FLUCTUATING TEMP
2060) C  DO NOT CHANGE THE 80,90 LOOP.
2061)     YY=Y-1
2062)     XX=X-1
2063)     DO 80 J=2,XX
2064)     DO 90 I=2,YY
2065)     RAY(I,J,2)=5
2066) 90    CONTINUE
2067) 80    CONTINUE
2068) C
2069) C  CONSTANT INTERIOR NODES.
2070) C  INDICATE RAY(I,J,2)=6 IF RELEVANT
2071) C
2072) C  IF RELEVANT, SPECIFY SPECIAL RAY(I,J,2) CONDITIONS
2073) C  TO BE USED WITH SEMI-INF BOUNDARY.
2074)     XX=X-1
2075)     YY=Y-1
2076)     DO 210 J=2,XX
2077)     RAY(YY,J,2)=12
2078) 210   CONTINUE
2079) C     RAY(YY,2,2)=15
2080)     RAY(YY,1,2)=23
2081) C     RAY(Y,2,2)=19
2082) C     RAY(YY,XX,2)=15
2083)     RAY(YY,X,2)=21
2084) C
2085) C  SET UP RAY(I,J,1) PRESENT NODAL TEMP
2086) C  DO NOT ALTER THIS
2087)     XX=X-1
2088)     YY=Y-1
2089)     DO 110 J=2,XX
2090)     RAY(1,J,1)=TSRF
2091)     RAY(Y,J,1)=TBTM
2092) 110   CONTINUE
2093)     DO 120 I=2,YY
2094)     RAY(I,1,1)=TLFT
2095)     RAY(I,X,1)=TRIT
2096) 120   CONTINUE
2097)     XX=X-1
2098)     YY=Y-1
2099)     DO 130 J=2,XX
2100)     DO 140 I=2,YY
2101) C  COMMENT OUT THE INITIAL TEMP DIST THAT YOU DON'T WANT:
2102) C  NODES ASSIGNED LINEAR TEMP DIST VERTICALLY
2103) C  RAY(I,J,1)=TSRF-((TSRF-TBTM)/(Y-1))*(I-1)
2104) C  NODES ASSIGNED UNIFORM TEMP (INDICATE THE TEMP:)
2105)     RAY(I,J,1)=4.67000
2106) 140   CONTINUE
2107) 130   CONTINUE
2108) C
2109) C  INTERIOR NODES
2110) C  INDICATE THE TEMPERATURES FOR RAY(I,J,1)
2111) C  IF THEY ARE DIFFERENT THAN ASSIGNED IN THE 130,140 LOOP.
2112) C
2113) C
2114) C  DO NOT CHANGE THE FOLLOWING 17 STATEMENTS.
2115)     WRITE(5,131) DS,DELT,DI,TDEL
2116) 131   FORMAT(1X,4F10.5)
2117)     WRITE(5,132) A,X,Y,NISO,ITRT,IMAX,ITPC
2118) 132   FORMAT(1X,7I5)
2119)     WRITE(5,136) ((ROPC(K),CPPC(K),HL(K),TPC(K)),K=1,A)
2120) 136   FORMAT(1X,4F10.5)
2121)     WRITE(5,134) (TISO(B),B=1,NISO)
2122) 134   FORMAT(1X,F7.2)
2123)     WRITE(5,133) (H(L),L=1,A)
2124) 133   FORMAT(1X,F10.5)
2125)     WRITE(5,151) ((RAY(I,J,3),J=1,X),I=1,Y)
2126)     WRITE(5,151) ((RAY(I,J,2),J=1,X),I=1,Y)
2127)     WRITE(5,151) ((RAY(I,J,1),J=1,X),I=1,Y)
2128)     WRITE(5,152) (KRNR(J),J=1,4)
2129) 152   FORMAT(1X,4I2)
2130) 151   FORMAT(1X,17F7.2)
2131)     WRITE(5,151) (FLXT(J),J=1,X)
2132)     WRITE(5,151) (FLXB(J),J=1,X)
2133)     WRITE(5,151) (FLXL(I),I=1,Y)
2134)     WRITE(5,151) (FLXR(I),I=1,Y)

```

```
2135)      WRITE(5,151) (TMPT(J),J=1,X)
2136)      WRITE(5,151) (TMPB(J),J=1,X)
2137)      WRITE(5,151) (TMPL(I),I=1,Y)
2138)      WRITE(5,151) (TMPR(I),I=1,Y)
2139)      RETURN
2140)      END
2141) C *****
```


File ADPOUT
(Lists the initial data in readable form and the
final values from the run [printed by ADIPC].)

DATA FOR THIS RUN OF ADIPC:

DS	DELT	DI	TDEL			
0.00500	0.00250	50.00000	1.00000			
A	X	Y	NISO	ITRT	IMAX	ITPC
1	3	25	1	40	1200	10
ROPC(K)	CPPC(K)	HL(K)	TPC(K)			
917.00000	93.87000	93.00000	0.00000			

TISO(B),B=1,NISO:
0.00

H(K)
0.00000

KRRR(1)= 1 KRRR(2)= 7 KRRR(3)= 7 KRRR(4)= 1

FLXT(J)	FLXB(J)	TMPT(J)	TMPB(J)
0.00	0.00	-4.67	4.67
0.00	0.00	-4.67	4.67
0.00	0.00	-4.67	4.67

FLXL(I)	FLXR(I)	TMPL(I)	TMPI(I)
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67
0.00	0.00	4.67	4.67

RAY(I,J,1) TEMPERATURES AT THE START:

-4.67	-4.67	-4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67
4.67	4.67	4.67

RAY(I,J,2) NODAL LOCATION TYPE

1.00	7.00	4.00
27.00	5.00	29.00
27.00	5.00	29.00
27.00	5.00	29.00
27.00	5.00	29.00

**Sample of isotherm locations output from ADIPC for the
Neuman problem (file POINT1)**

```

THE FOLLOWING      3 POINTS REPRESENT TIME=  0.0250 HOURS:
  3      1  0.00
    0.000000      0.004632
    0.005000      0.004632
    0.010000      0.004632
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.0500 HOURS:
  3      1  0.00
    0.000000      0.004727
    0.005000      0.004727
    0.010000      0.004727
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.0750 HOURS:
  3      1  0.00
    0.000000      0.004830
    0.005000      0.004830
    0.010000      0.004830
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.1000 HOURS:
  3      1  0.00
    0.000000      0.004940
    0.005000      0.004940
    0.010000      0.004940
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.1250 HOURS:
  3      1  0.00
    0.000000      0.005197
    0.005000      0.005197
    0.010000      0.005197
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.1500 HOURS:
  3      1  0.00
    0.000000      0.005661
    0.005000      0.005661
    0.010000      0.005661
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.1750 HOURS:
  3      1  0.00
    0.000000      0.006161
    0.005000      0.006161
    0.010000      0.006161
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.2000 HOURS:
  3      1  0.00
    0.000000      0.006685
    0.005000      0.006685
    0.010000      0.006685
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.2250 HOURS:
  3      1  0.00
    0.000000      0.007223
    0.005000      0.007223
    0.010000      0.007223
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.2500 HOURS:
  3      1  0.00
    0.000000      0.009329
    0.005000      0.009329
    0.010000      0.009329
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.2750 HOURS:
  3      1  0.00
    0.000000      0.009413
    0.005000      0.009413
    0.010000      0.009413
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.3000 HOURS:
  3      1  0.00
    0.000000      0.009514
    0.005000      0.009514
    0.010000      0.009514
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.3250 HOURS:
  3      1  0.00
    0.000000      0.009618
    0.005000      0.009618
    0.010000      0.009618
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.3500 HOURS:
  3      1  0.00
    0.000000      0.009724
    0.005000      0.009724
    0.010000      0.009724
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.3750 HOURS:
  3      1  0.00
    0.000000      0.009831
    0.005000      0.009831
    0.010000      0.009831
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.4000 HOURS:
  3      1  0.00
    0.000000      0.009941
    0.005000      0.009941
    0.010000      0.009941
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.4250 HOURS:
  3      1  0.00
    0.000000      0.010158
    0.005000      0.010158
    0.010000      0.010158

```