70-112

GEOLOGICAL SURVEY
COMPUTER CONTRIBUTION

NUMBER 4

Hot Pipe

by

Patrick C. Doherty

Computer Center Division
Menlo Park, California
1970

Program Number:	C102
Sponsor:	Arthur H. Lachenbruch
Equipment:	IBM System 360/65
Operating System:	IBM System 360
Language:	PL/1

Open File Report

COMPUTER CONTRIBUTION

1. Weighted Triangulation Adjustment, by Walter L. Anderson, 1969
2. Perspective Center Determination, by John D. McLaurin, 1969
3. Non-Constant Variance Regression Analysis, by Marshall Strong Hellmann, 1970
4. Hot Pipe, by Patrick C. Doherty, 1970

Free on application to the Chief, Computer Center Division,
U. S. Geological Survey, Washington, D. C. 20242

CONTENTS

ILLUSTRATIONS

HOT PIPE

by Patrick C. Doherty

ABSTRACT

This program computes the temperature history of a cross-section of permafrost in which an oil pipe is buried in a ten-foot-deep trench. The surface of the pipe is assumed to be maintained at a constant temperature of 80 degrees centigrade. The surface of the permafrost is subject to seasonal variation in temperature. The thermal properties of frozen and thawed material are treated, as well as the latent heat of melting.

The equation of heat conduction in two-dimensional cartesian coordinates is solved by the alternating direction explicit method to obtain the temperature history. Output is in the form of printer plots of isothermal lines as well as complete numeric printouts of the temperature array and the internal energy array.

USE AND INTRODUCTION

This program was written in response to a request for information on the possible effects that construction of the proposed pipeline across Alaska would have on the natural environment. The results of this thermal analysis form the basis for a study of such effects.

The program was written with the cooperation of Arthur H. Lachenbruch, who was responsible for the input data, the physical efficacy of the model, and the physical interpretation of computed results.

TECHNICAL DESCRIPTION

The cross-section of permafrost which is our problem space is covered with a grid as shown in figure 1. The left boundary of this grid passes through the center of the pipe. Since the solution is symmetrical about a vertical line passing through the center of the pipe, only the right half of the solution need be computed. The grid lines are two feet apart and cover a rectangle sixty feet wide and ninety feet deep. The pipe is represented by a four foot square, the right half of which is shown by the darkened area in the upper left corner of the grid.

We calculate the temperature at each of the intersection points of the grid. Each of these points may be thought of as the center point of a square "cell" whose sides are drawn midway between the grid points. The temperature at each grid point may be considered the average temperature of the material in the cell. A small part of the grid showing temperatures defined at the centers of cells is shown in figure 2. This figure also illustrates the indexing scheme that we will use in our analysis.

The two-dimensional equation of heat conduction for a non-moving medium without sources is:

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

If ρ, c, and k are constants, then:

$$\rho C \frac{\partial T}{\partial t} = k\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right). \quad (2)$$

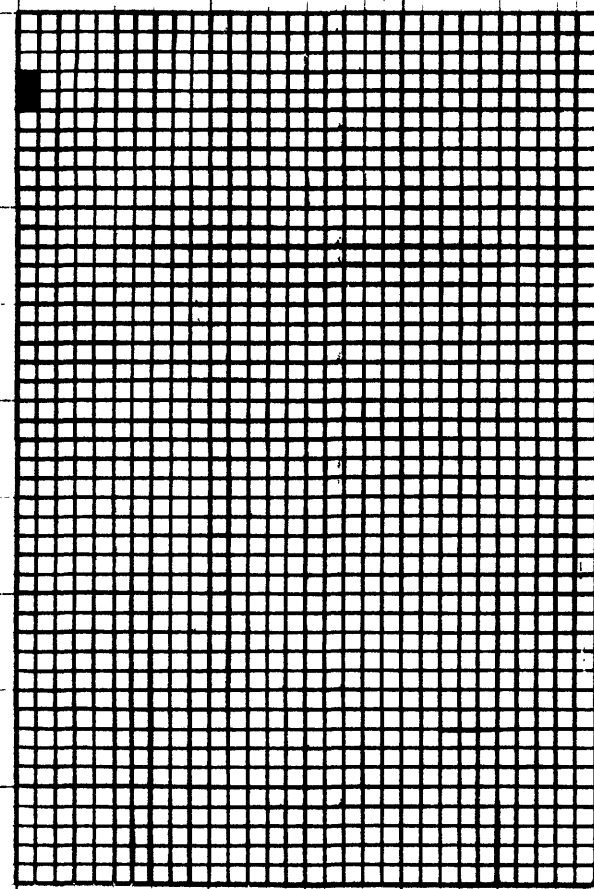


Figure 1.--Cross-section of permafrost.

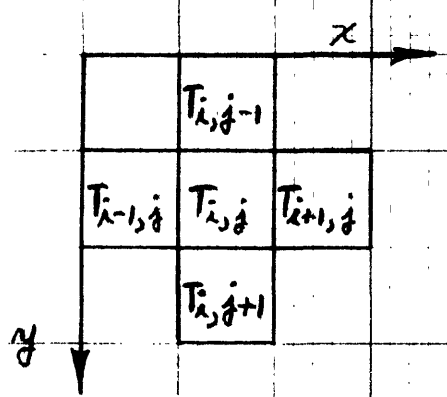


Figure 2.--Temperatures defined at the centers of cells.

Guided by the indexing scheme shown in figure 2 above, the partial derivatives in space may be approximated by finite differences as follows:

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\Delta x^2}\left[(T_{i+1,j} - T_{i,j}) + (T_{i-1,j} - T_{i,j})\right]; \tag{3}$$

$$\frac{\partial^2 T}{\partial y^2} = \frac{1}{\Delta y^2}\left[(T_{i,j+1} - T_{i,j}) + (T_{i,j-1} - T_{i,j})\right]. \tag{4}$$

Substituting (3) and (4) into (2) we obtain:

$$\rho C \frac{\partial T}{\partial t} = \frac{k}{\Delta x^2}\left[(T_{i+1,j} - T_{i,j}) + (T_{i-1,j} - T_{i,j})\right] + \frac{k}{\Delta y^2}\left[(T_{i,j+1} - T_{i,j}) + (T_{i,j-1} - T_{i,j})\right]. \tag{5}$$

Equation (5) is incomplete because we have not specified the value of time at which the space derivatives are computed. We want the time derivative, $\partial T/\partial t$, to be the average time derivative between two time slices, so that the finite-difference approximation for the time derivative will be more accurate. Hence, in (5) we want to put some average of the space derivatives at the two time slices. If we indicate two time slices by the superscripts n and n+1, then instead of (5) we want something like the following:

$$\rho C \frac{\partial T}{\partial t} = \beta^{n+1} k \left[\frac{(T^{n+1}_{i+1,j} - T^{n+1}_{i,j}) + (T^{n+1}_{i-1,j} - T^{n+1}_{i,j})}{\Delta x^2} + \frac{(T^{n+1}_{i,j+1} - T^{n+1}_{i,j}) + (T^{n+1}_{i,j-1} - T^{n+1}_{i,j})}{\Delta y^2}\right] + \beta^{n} k \left[\frac{(T^{n}_{i+1,j} - T^{n}_{i,j}) + (T^{n}_{i-1,j} - T^{n}_{i,j})}{\Delta x^2} + \frac{(T^{n}_{i,j+1} - T^{n}_{i,j}) + (T^{n}_{i,j-1} - T^{n}_{i,j})}{\Delta y^2}\right]. \tag{6}$$

If we assign the value 1/2 to each of the weights, β^{n+1} and β^{n}, we would be doing a simple averaging of the space derivatives at the two time slices. This simple, intuitively appealing strategy gives us the Crank-Nicolson formula, which was one of the first integration formulas to be used in solving diffusion problems.

It is less intuitive to apply separate weighting factors to each of the terms contained in parentheses on the right-hand side of (6), but this strategy leads to the more useful alternating-direction implicit method, and the alternating-direction explicit method. Our choice is the alternating-direction explicit (ADE) method.

Substituting the finite-difference approximation

$$\frac{\partial T}{\partial t} = \frac{(T_{i,j}^{n+1} - T_{i,j}^{n})}{\Delta t}$$

into (6), and applying separate weights to each term we obtain the following generalized formula:

$$\begin{aligned}
\rho C \frac{(T_{i,j}^{n+1} - T_{i,j}^{n})}{\Delta t} = {} & \beta_1 k \frac{(T_{i+1,j}^{n+1} - T_{i,j}^{n+1})}{\Delta x^2} \\
& + \beta_2 k \frac{(T_{i+1,j}^{n} - T_{i,j}^{n})}{\Delta x^2} \\
& + \beta_3 k \frac{(T_{i-1,j}^{n+1} - T_{i,j}^{n+1})}{\Delta x^2} \\
& + \beta_4 k \frac{(T_{i-1,j}^{n} - T_{i,j}^{n})}{\Delta x^2} \\
& + \beta_5 k \frac{(T_{i,j+1}^{n+1} - T_{i,j}^{n+1})}{\Delta y^2} \\
& + \beta_6 k \frac{(T_{i,j+1}^{n} - T_{i,j}^{n})}{\Delta y^2} \\
& + \beta_7 k \frac{(T_{i,j-1}^{n+1} - T_{i,j}^{n+1})}{\Delta y^2} \\
& + \beta_8 k \frac{(T_{i,j-1}^{n} - T_{i,j}^{n})}{\Delta y^2} .
\end{aligned} \tag{7}$$

Equation (7) above is a general form from which all of the current numerical methods for solving the heat equation are derived by the manner in which the weights, βi, are assigned.

In the alternating-direction explicit method, two different complementary formulas are derived from (7), and the numerical integration proceeds forward in time by half-steps, using the two formulas alternately to sweep over the solution grid from the northwest corner to the southeast corner, and then back from the southeast corner to the northwest corner.

For the sweep from northwest to southeast, the following weights are assigned to the β's in equation (7):

$$\begin{aligned} \beta_1 = \beta_4 = \beta_5 = \beta_8 = 0 \\ \beta_2 = \beta_3 = \beta_6 = \beta_7 = 1. \end{aligned} \tag{8}$$

This assignment of weights gives the formula:

$$\begin{aligned} \rho C \frac{(T_{i,j}^{n+1} - T_{i,j}^{n})}{\Delta t} = k \frac{(T_{i+1,j}^{n} - T_{i,j}^{n})}{\Delta x^2} \\ + k \frac{(T_{i-1,j}^{n+1} - T_{i,j}^{n+1})}{\Delta x^2} \\ + k \frac{(T_{i,j+1}^{n} - T_{i,j}^{n})}{\Delta y^2} \\ + k \frac{(T_{i,j-1}^{n+1} - T_{i,j}^{n+1})}{\Delta y^2}. \end{aligned} \tag{9}$$

Solving (9) for $T_{i,j}^{n+1}$ gives the following:

$$T_{i,j}^{n+1} = \frac{\left[\rho C \frac{T_{i,j}^{n}}{\Delta t} + k\frac{(T_{i+1,j}^{n} - T_{i,j}^{n})}{\Delta x^2} + k\frac{T_{i-1,j}^{n+1}}{\Delta x^2} + k\frac{(T_{i,j+1}^{n} - T_{i,j}^{n})}{\Delta y^2} + k\frac{T_{i,j-1}^{n+1}}{\Delta y^2}\right]}{\left[\frac{\rho C}{\Delta t} + \frac{k}{\Delta x^2} + \frac{k}{\Delta y^2}\right]}. \quad (10)$$

In formula (10) above and in other formulas to be derived, the value ρC is taken to be that appropriate to the grid point where $T_{i,j}^{n+1}$ is being computed. The value of k is taken to be that appropriate to the cell boundary separating the two grid points between which temperature differences are being taken. Hence, in (10), k does not stand for a single number but for four different numbers corresponding to the four sides of the cell. In cases where the grid spacing is variable, the same considerations apply to the quantities represented by Δx and Δy.

On the western boundary of the problem where $\partial T/\partial x = 0$, the second term on the right-hand side of (9) is zero. In this case, (10) reduces to the following:

$$T_{i,j}^{n+1} = \frac{\left[\rho C \frac{T_{i,j}^{n}}{\Delta t} + k\frac{(T_{i+1,j}^{n} - T_{i,j}^{n})}{\Delta x^2} + k\frac{(T_{i,j+1}^{n} - T_{i,j}^{n})}{\Delta y^2} + k\frac{T_{i,j-1}^{n+1}}{\Delta y^2}\right]}{\left[\frac{\rho C}{\Delta t} + \frac{k}{\Delta y^2}\right]}. \quad (11)$$

Similarly, on the eastern boundary where also $\partial T/\partial x = 0$, equation (10) reduces to:

$$T_{i,j}^{n+1} = \frac{\left[\rho C \frac{T_{i,j}^{n}}{\Delta t} + k\frac{T_{i-1,j}^{n+1}}{\Delta x^2} + k\frac{(T_{i,j+1}^{n} - T_{i,j}^{n})}{\Delta y^2} + k\frac{T_{i,j-1}^{n+1}}{\Delta y^2}\right]}{\left[\frac{\rho C}{\Delta t} + \frac{k}{\Delta x^2} + \frac{k}{\Delta y^2}\right]}. \tag{12}$$

For the sweep from southeast to northwest, the following values are assigned to the β's in equation (7):

$$\begin{aligned} &\beta_1 = \beta_4 = \beta_5 = \beta_8 = 1 \\ &\beta_2 = \beta_3 = \beta_6 = \beta_7 = 0. \end{aligned} \tag{13}$$

This assignment of weights gives the formula:

$$\begin{aligned} \rho C \frac{(T_{i,j}^{n+1} - T_{i,j}^{n})}{\Delta t} &= k\frac{(T_{i+1,j}^{n+1} - T_{i,j}^{n+1})}{\Delta x^2} \\ &+ k\frac{(T_{i-1,j}^{n} - T_{i,j}^{n})}{\Delta x^2} \\ &+ k\frac{(T_{i,j+1}^{n+1} - T_{i,j}^{n+1})}{\Delta y^2} \\ &+ k\frac{(T_{i,j-1}^{n} - T_{i,j}^{n})}{\Delta y^2}. \end{aligned} \tag{14}$$

Solving (14) for $T_{i,j}^{n+1}$ gives the formula:

$$T_{i,j}^{n+1} = \frac{\left[\rho C \frac{T_{i,j}^{n}}{\Delta t} + k \frac{T_{i+1,j}^{n+1}}{\Delta x^2} + k \frac{(T_{i-1,j}^{n} - T_{i,j}^{n})}{\Delta x^2} + k \frac{T_{i,j+1}^{n+1}}{\Delta y^2} + k \frac{(T_{i,j-1}^{n} - T_{i,j}^{n})}{\Delta y^2} \right]}{\left[\frac{\rho C}{\Delta t} + \frac{k}{\Delta x^2} + \frac{k}{\Delta y^2} \right]} \quad (15)$$

On the western boundary, (15) reduces to the following:

$$T_{i,j}^{n+1} = \frac{\left[\rho C \frac{T_{i,j}^{n}}{\Delta t} + k \frac{T_{i+1,j}^{n+1}}{\Delta x^2} + k \frac{T_{i,j+1}^{n+1}}{\Delta y^2} + k \frac{(T_{i,j-1}^{n} - T_{i,j}^{n})}{\Delta y^2} \right]}{\left[\frac{\rho C}{\Delta t} + \frac{k}{\Delta x^2} + \frac{k}{\Delta y^2} \right]} \quad (16)$$

On the eastern boundary, (15) reduces to the following:

$$T_{i,j}^{n+1} = \frac{\left[\rho C \frac{T_{i,j}^{n}}{\Delta t} + k \frac{(T_{i-1,j}^{n} - T_{i,j}^{n})}{\Delta x^2} + k \frac{T_{i,j+1}^{n+1}}{\Delta y^2} + k \frac{(T_{i,j-1}^{n} - T_{i,j}^{n})}{\Delta y^2} \right]}{\left[\frac{\rho C}{\Delta t} + \frac{k}{\Delta y^2} \right]} \quad (17)$$

Formulas (10), (11), (12), (15), (16), and (17) constitute our basic solution algorithm.

Management of Melting

Figure 3 illustrates the traditional approach to the heat conduction problem in two materials. If one began the solution in Material I, then the differential equation $\rho C_I \partial T/\partial t = k_I(\partial^2 T/\partial x^2 + \partial^2 T/\partial y^2)$ would be used to march the solution throughout region I. The boundary condition $k_I(\partial T/\partial y)_I = k_{II}(\partial T/\partial y)_{II}$ would be used to march the solution across the interface between the two materials, and finally the equation $\rho C_{II} \partial T/\partial t = k_{II}(\partial^2 T/\partial x^2 + \partial^2 T/\partial y^2)$ would be used to march the solution throughout region II.

The boundary condition $k_I(\partial T/\partial y)_I = k_{II}(\partial T/\partial y)_{II}$ expresses the idea that the number of calories of heat that flow into the boundary from one material must flow out of the boundary into the other material. There are no calories produced or consumed at the boundary.

In the case where the two materials in question are the liquid and solid phases of the same material, then calories are produced and consumed at the liquid-solid interface, and the boundary condition applied above is not valid. In this case, however, the temperature of the boundary is a known fixed quantity, namely the melting-freezing temperature of the material.

We propose to represent the liquid-solid interface by a series of connected cells where the temperature is passing through the melting-freezing temperature of the material, as shown in figure 4. The onset of melting in any cell is indicated when the temperatures at two successive time slices, $T^n_{i,j}$ and $T^{n+1}_{i,j}$ straddle the melting temperature of the material, T_M. When the onset of melting is detected, the cell temperature is set to T_M and the calorie inflow indicated by the temperature difference $T^{n+1}_{i,j} - T_M$ is calculated and accumulated into a variable, $E_{i,j}$, representing the increase in the internal energy of the material contained in the cell.

The volume of the cell is $1 \times (\Delta x \Delta y)$. Its mass is $\rho \Delta x \Delta y$. If P is the fraction of meltable material in the cell, and L is the calories required to melt one unit mass of meltable material, then the total calories required to melt all meltable material in the cell is $\rho \Delta x \Delta y P L$.

If C_F is the specific heat of the frozen material, then the calorie input required to produce the temperature difference $T^{n+1}_{i,j} - T_M$ is given by:

$$Q^{n+1} = (T^{n+1}_{i,j} - T_M) C_F \rho \Delta x \Delta y \qquad (18)$$

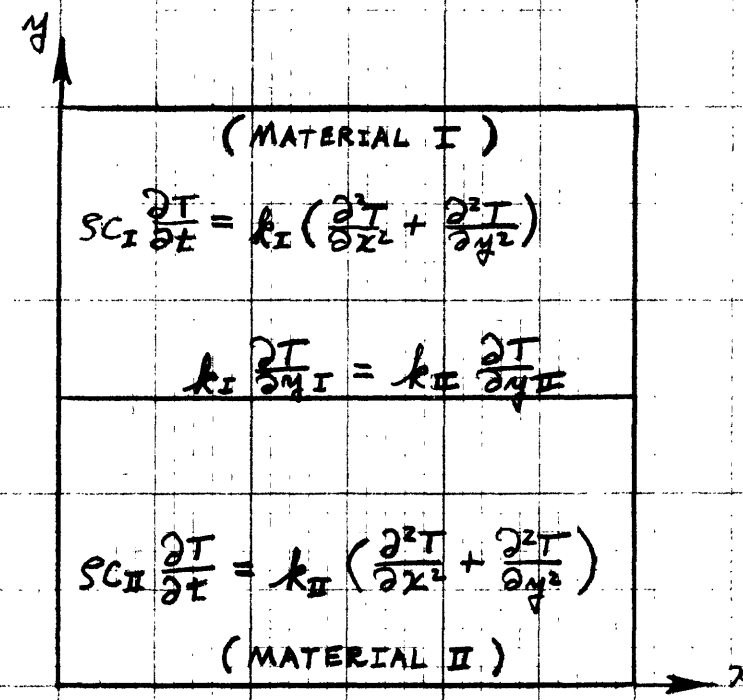


Figure 3.--Traditional approach to the heat conduction problem in two materials and across the interface between the two materials.

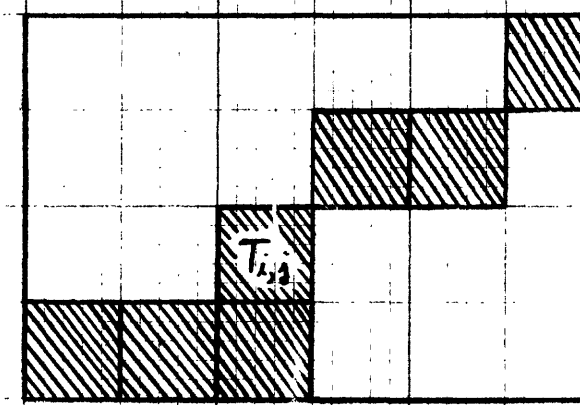


Figure 4.--Representation of liquid-interface by a series of connected cells.

In succeeding time slices, the no-melting temperature change is computed in the usual way, but the temperature of the cell is retained at T_M and the calorie inflow corresponding to the no-melting temperature change is added to the internal energy of the cell:

$$Q^{n+2} = (T_{i,j}^{n+2} - T_M) C_F \rho \Delta x \Delta y$$
$$Q^{n+3} = (T_{i,j}^{n+3} - T_M) C_F \rho \Delta x \Delta y \qquad (19)$$
$$\text{etc.}$$

$$E_{i,j}^{n+m} = \sum_{l=1}^{m} Q^{n+l}. \qquad (20)$$

When the internal energy of the cell, $E_{i,j}$, equals or exceeds the energy required to melt the meltable material in the cell, then the cell is considered to have changed from solid to liquid:

$$E_{i,j}^{n+m} \geq Q_n = \rho \Delta x \Delta y P L. \qquad (21)$$

Any excess calories that are accumulated in the time slice where the solid-to-liquid transition takes place are applied to raising the temperature of the liquid. If C_T is the specific heat of the thawed material then:

$$E_{i,j}^{n+m} - Q_n = (T_{i,j}^{n+m} - T_M) C_T \rho \Delta x \Delta y, \qquad (22)$$

where

$$T_{i,j}^{n+m} = T_M + \frac{E_{i,j}^{n+m} - Q_n}{C_T \rho \Delta x \Delta y}.$$

The reverse procedure to that outlined above would apply when the temperature was dropping, and freezing of the liquid was taking place.

D. Quon and others (1965) compare the ADE method with other methods and include references to the papers by Saul'ev and Larkin.

Truncation Error

In order to get some feeling for the truncation error related to the finite time step, Δt , and its relation to the grid intervals, Δx and Δy , we have examined three specific examples as shown in figures 5, 6, and 7. In figure 5, we have a square grid with $\Delta x = \Delta y = 15$. The middle temperature $T_A = 0$, and the four neighboring temperatures $T_B = -2$, $T_C = +3$, $T_D = +5$, $T_E = -1$. In figure 6 we have another square grid with the grid interval increased by a factor of 4; $\Delta x = \Delta y = 60$. The middle temperature remains the same $T_A = 0$, but the four neighboring temperatures are multiplied by 4 so that slopes will remain the same; $T_B = -8$, $T_C = +12$, $T_D = +20$ $T_E = -4$. In figure 7 we have a rectangular grid with Δy , T_C , and T_E taken from figure 5, and Δx , T_B , and T_D taken from figure 6. In all of these examples we set $\alpha = k/\rho C = 0.006$. For each of these three examples we will examine the northwest-to-southeast formula given in equation (9):

$$\frac{T_A^* - T_A}{\Delta t} = \frac{\alpha}{\Delta x^2}\left[(T_B - T_A) + (T_C^* - T_A^*) + (T_D^* - T_A^*) + (T_E - T_A)\right]. \qquad (23)$$

In equation (23) above, the starred temperatures are new temperatures corresponding to current time while unstarred temperatures are old temperatures corresponding to the previous time slice.

At equilibrium, $T_A^* = T_A$, $(T_A^* - T_A)/\Delta t = 0$, and (23) reduces to the familiar:

$$T_A = \frac{1}{4}(T_B + T_C^* + T_D^* + T_E^*). \qquad (24)$$

Formula (24) gives equilibrium temperature of 1.25 in the case where $\Delta x = \Delta y = 15$, and the equilibrium temperature of 5.0 in the case where $\Delta x = \Delta y = 60$.

Solving (23) for T_A^* we obtain:

$$T_A^* = \frac{\frac{T_A}{\Delta t} + \frac{\alpha}{\Delta x^2}(T_B + T_C^* + T_D^* + T_E - 2T_A)}{\left(\frac{1}{\Delta t} + \frac{2\alpha}{\Delta x^2}\right)}. \qquad (25)$$

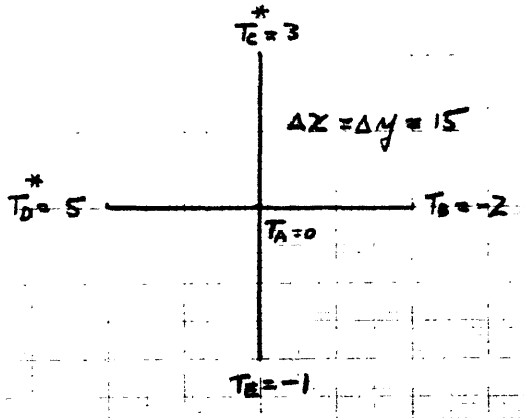


Figure 5.--Example of a square grid.

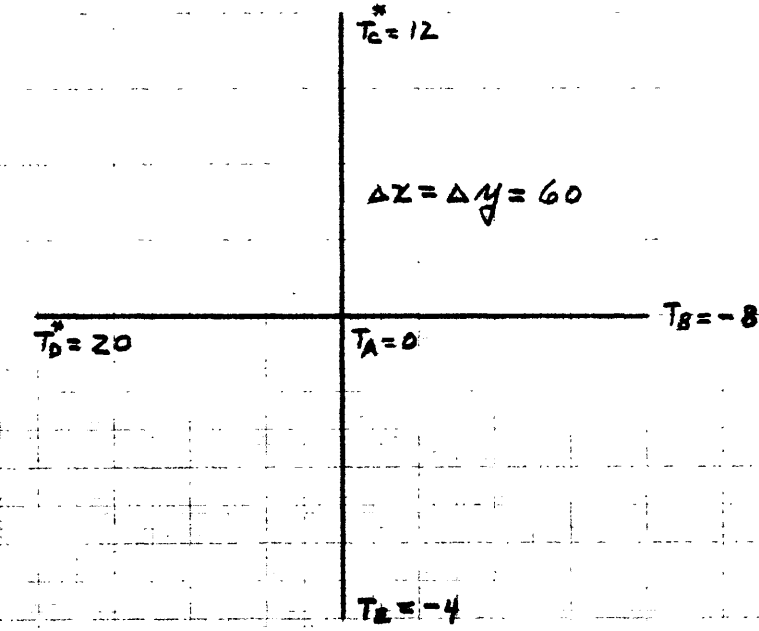


Figure 6.--Example of another square grid.

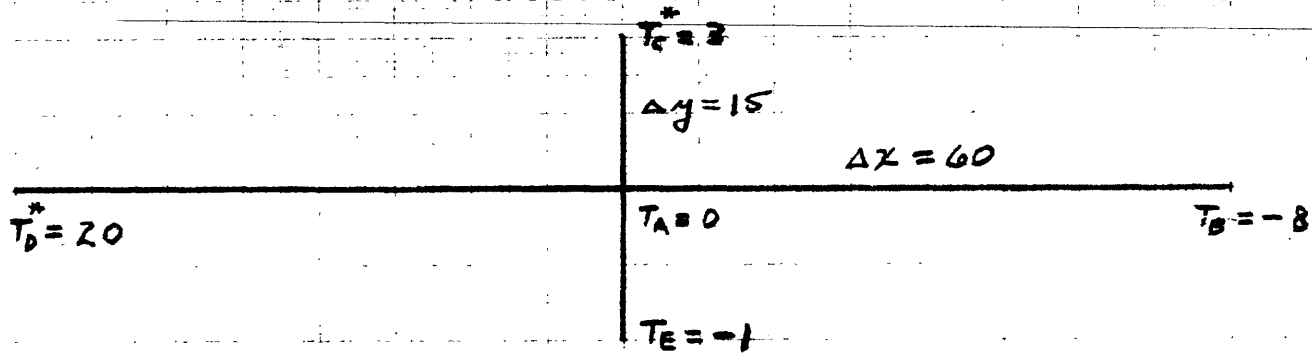


Figure 7.--Example of a rectangular grid.

If we permit $\Delta t \to \infty$ in equation (25) we obtain the limiting value of T_A^* as follows:

$$T_A^* = \frac{1}{2}\left(T_B + T_C^* + T_D^* + T_E - 2T_A\right). \tag{26}$$

Formula (26) gives limiting temperature $T_A^*=2.5$ for the case where $\Delta x = \Delta y = 15$, and limiting temperature $T_A^* = 10.0$ for the case where $\Delta x = \Delta y = 60$.

Substituting the numbers from example 5 and example 6 into formula (25) we get respectively:

$$T_A^* = \frac{1.333 \times 10^{-4}}{\frac{1}{\Delta t} + 5.333 \times 10^{-5}}; \tag{27}$$

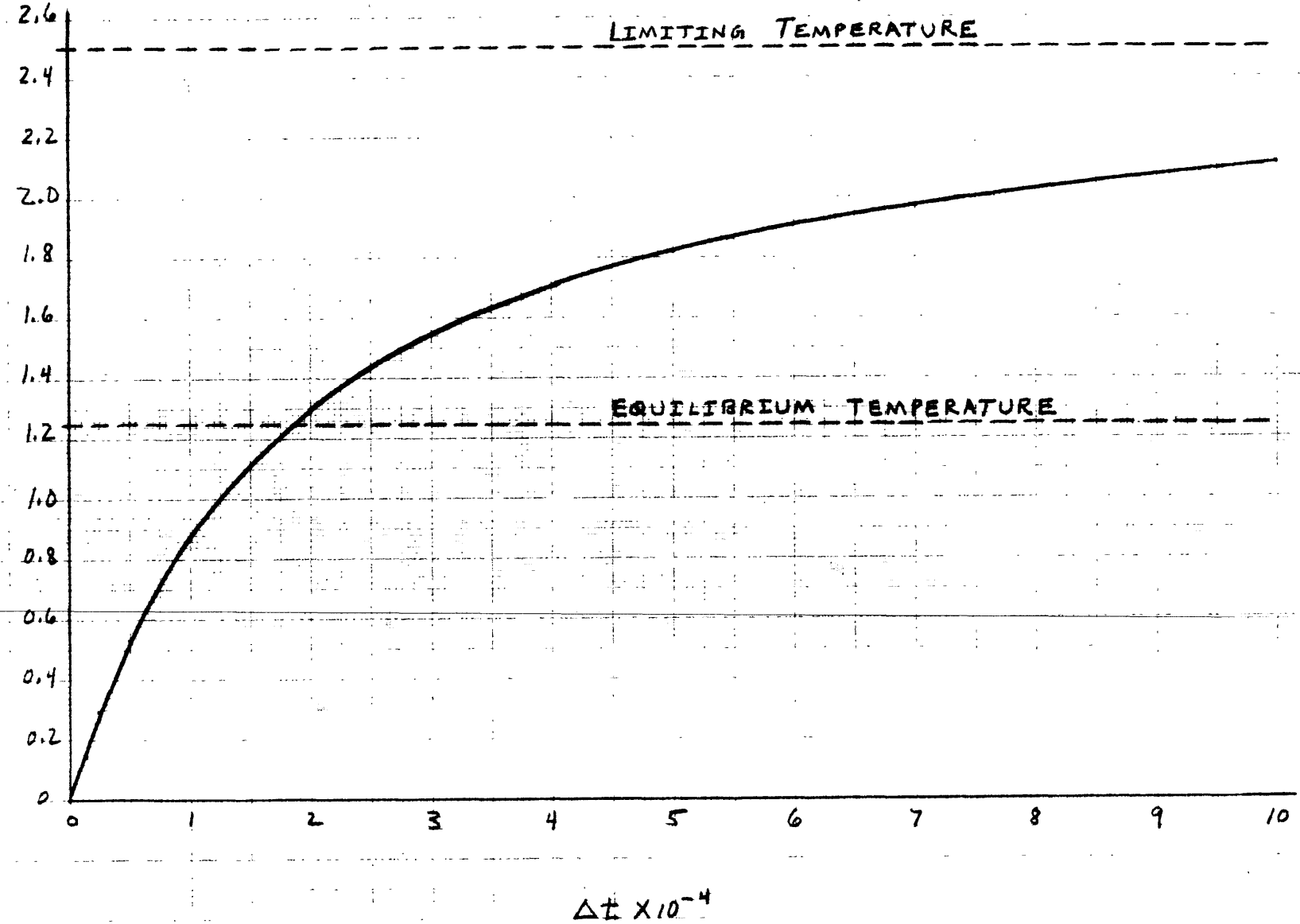
and,

$$T_A^* = \frac{8.333 \times 10^{-5}}{\frac{1}{\Delta t} + 3.333 \times 10^{-6}}. \tag{28}$$

Plots of T_A^* vs. Δt for these two examples are shown in figure 8 and figure 9. These figures show how the curve of T_A^* vs. Δt starts from the origin in a reasonably linear fashion but soon curves over and approaches the limiting temperature in an asymptotic manner. Furthermore, the asymptotic temperature is well above the equilibrium temperature which is crossed rather early in the trace of T_A^*. These two facts relate to two serious pitfalls that result from an excessively large time step.

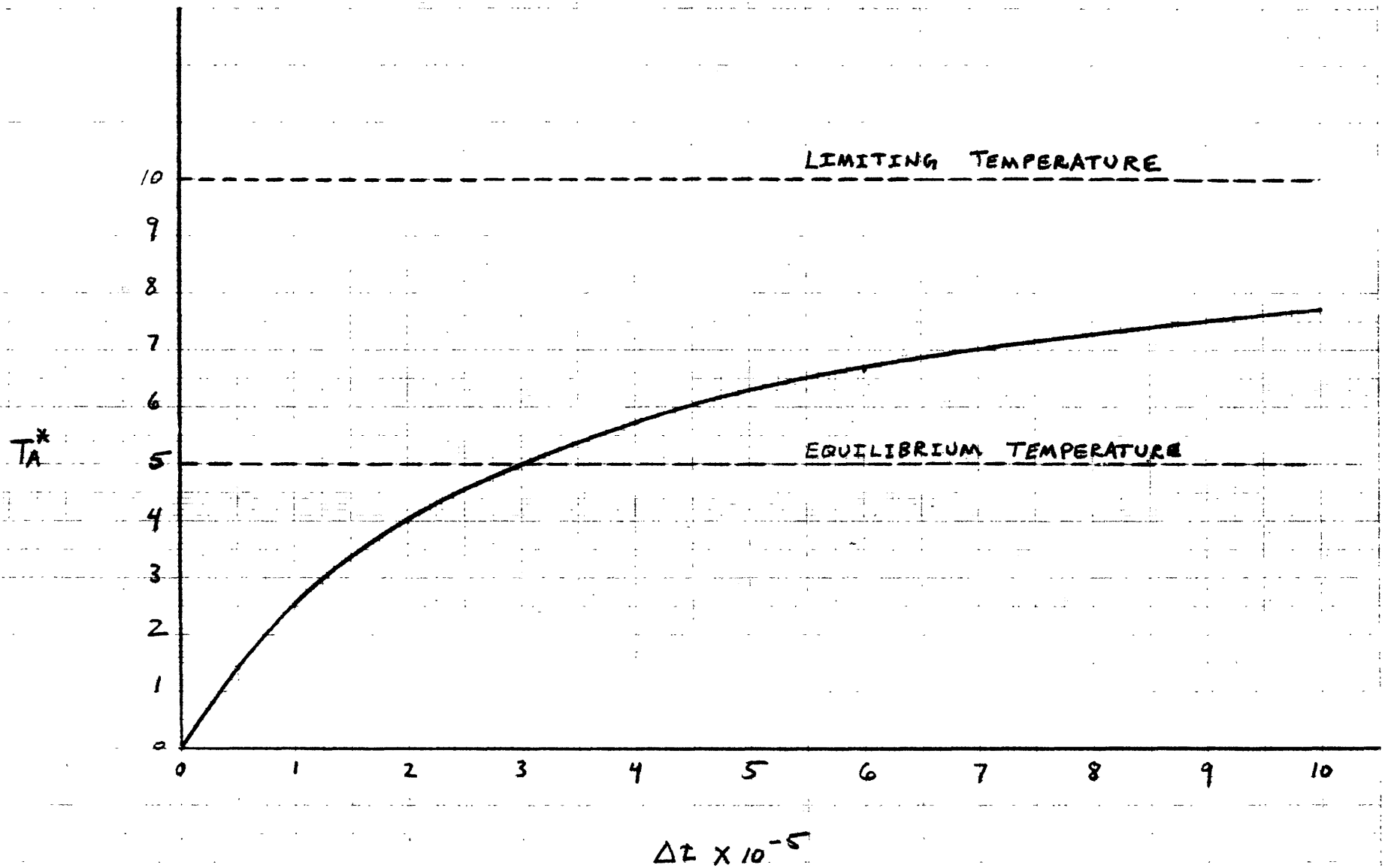
First of all, if the calculated temperature, T_A^*, is being used to calculate the inflow of calories into a cell undergoing melting during the time interval, Δt, it is clear that there is a definite limit to the calorie inflow per time step regardless of the size of Δt. Hence, if Δt is so large as to depart from the relatively linear portion of the curve, the melting rate will seem to slow down, and the temperature history will be retarded.

Secondly, if Δt is large enough, the calculated temperature, T_A^*, may exceed the equilibrium temperature, thus doing violence to the basic physical principles involved in the problem, and inducing a kind of pathological oscillation in the temperature history.



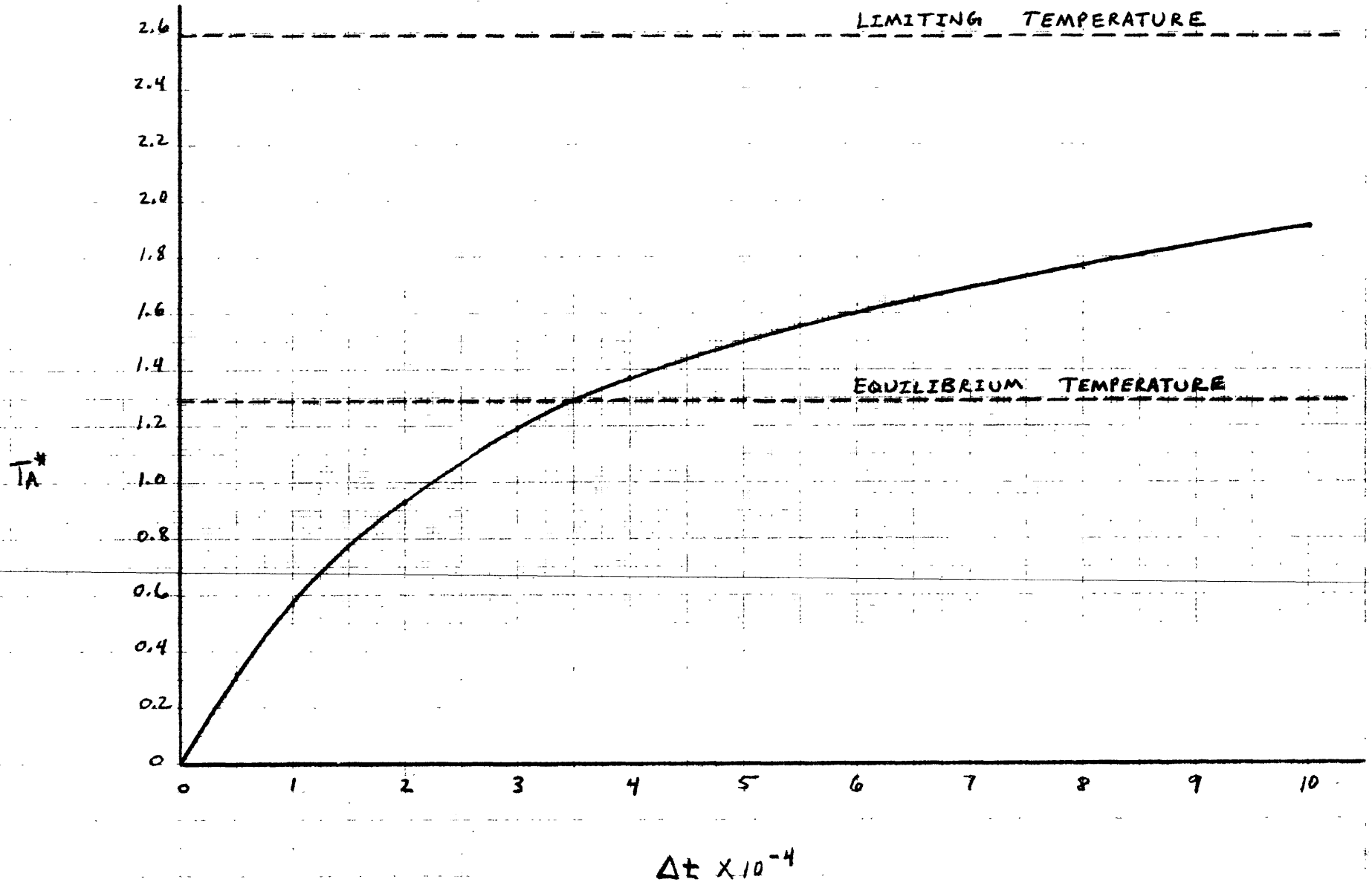
$\Delta x = 15$
$\alpha = 0.006$

Figure 8 - T_A^* vs. Δt for figure 5.



$\Delta x = 60$
$\alpha = 0.006$

Figure 9 - T_A^* vs. Δt for figure 6.



$\Delta x = 60$, $\Delta y = 15$

$\alpha = 0.006$

Figure 10 - T_A^* vs. Δt for figure 7.

It is clear that one must choose a time step that will not cause the equilibrium temperature to be passed, and furthermore one must remain on the relatively linear portion of the curve of T_A^* vs. Δt. With this rule in mind, we see from figure 8 that for the case where the grid interval $\Delta x = \Delta y = 16$, our time step is limited to about 10^4, whereas from figure 9 we see that increasing the grid interval by a factor of four permits an increase in the time step to about 10^5.

This shows that our resolution in time is related to our resolution in space. We cannot increase our resolution in space indefinitely without also increasing our resolution in time. Or to put it another way, if we wish no more than a certain minimum resolution in time, we must relax our resolution in space to the point where the two are compatible. Figure 10 shows that when we have a rectangular rather than a square grid, our time step is pretty much limited by the smaller of the two grid intervals.

The preparation of curves such as those given in figures 8, 9, and 10 is a useful preliminary step when one is about to embark on a calculation using ADE (or any other method). Such specific examples will give an approximate indication of the maximum time step that may be used safely. However, since there is no general formula for truncation error, one must always perform some numerical experiments, calculating time histories at the indicated step size and again at half that step size and comparing numerical results to confirm that the step size will give acceptable accuracy.

PROGRAM DESCRIPTION

The program is organized in the form of a dummy main program called HOTPIPE, the true main control block called CONTROL, and a multitude of closed subroutines nested within CONTROL.

The principal functions of the main program, HOTPIPE, are to contain declarations for all variables and arrays that are not dynamically allocated, to read in basic parameters from cards and calculate computed constants, and finally to print a title page and problem-identifying parameters.

All problem variables are declared either fixed binary (31) or floating decimal (16). The linesize for the file SYSPRINT is expanded to 131 columns from 120 to permit wider printer plots and to print 10 floating point numbers per line. In the statement labeled READIN we cause program exit when the end-of-file is found in SYSIN. This permits data for several problems to be stacked in SYSIN. After input of all

basic parameters, including grid dimensions NROW and NCOL, computed constants are obtained, including indexes and line images needed for the printer plots. Finally the title page and problem-identifying parameters are printed and the main control block is entered at statement CONTROL.

The NOUNDERFLOW condition is set, upon entry to CONTROL, to prevent printing of diagnostics when underflow occurs. The temperature array and internal energy array are declared using the dimensions NROW and NCOL, previously read from data cards. The "ON" statement sets up a transfer to the error routine BUGOUT if an overflow is detected in subsequent calculations. Then time, step counters, and the ambient temperature profiles are initialized. Pipe temperatures and internal energy profiles are initialized. A print plot of initial conditions is printed and the initial temperature array is printed in full. Finally the forward integration begins with the statement labeled LOOP.

At the beginning of the integration loop, step counters are advanced, time is advanced, and the surface boundary temperature is advanced. Then the downward sweep of the ADE is performed, the surface temperature is advanced again, and the upward sweep of the ADE is performed. This completes one forward integration step. If a print point has been reached, a printer plot is produced. If a stop point has not been reached, control then returns to the statement LOOP for another forward integration step.

When a stop point is reached, the temperature array, T, and the internal energy array, E, are printed in full. After these arrays are printed, new integration parameters are read from data cards. Any parameter may be changed at this time, but the intent of this input is to permit changing the integration time step and the frequency of print-out. Time step and print frequency are controlled by the parameters NTIME, NPRINT, and NSTOP. NTIME is the number of print points per year, NPRINT is the number of integration steps between print points, and NSTOP is the number of prints before the next stopping point. The new integration parameters are printed, and control returns to the statement LOOP to continue the forward integration. If the parameter NTIME is set to zero, this indicates the end of one time history. Control then goes to the statement READIN to permit entry of data for a new history.

The two major subroutines called within the forward integration loop are DNSWEEP and UPSWEEP which control the downward and upward sweeps of the alternating-direction explicit algorithm. The main function of these two subroutines is to control the calculation around the pipe boundary which is the only irregularity in an otherwise simple rectangular boundary. The center of the pipe is placed at a grid point which is determined by the input parameter NCL. This permits varying the depth of the pipe in units of one grid interval. These two subroutines call six computational subroutines for the calculation of temperatures.

Subroutines FORMA, FORMB, and FORMC are called by DNSWEEP to compute new temperatures at the left boundary, the interior, and the right boundary of the grid respectively. The formulas applied are based on equation (9) for the sweep from northwest to southeast. Subroutines FORMD, FORME, and FORMF are called by UPSWEEP to compute new temperatures at the right boundary, the interior, and the left boundary of the grid respectively. The formulas are based on equation (14) for the sweep from southeast to northwest. At the top and bottom boundaries, the temperature is specified; fixed at the bottom and varying with time at the top. At the left and right boundaries, we specify that $\partial T/\partial x = 0$, except at the pipe where a fixed pipe temperature is specified. The six computational subroutines are identical except for the formula used to compute the new value of temperature. Each subroutine calls subroutine ALPHA for conductivities, calculates the new temperature value, and then calls subroutine MELFRZ for the management of melting or freezing.

Subroutine ALPHA gets its name from the fact that the program was originally written to use diffusivities, and was later changed so that conductivities were separated from density and specific heat. This is because conductivities are evaluated at cell boundaries and density times specific heat is evaluated at cell centers. The name of the subroutine was retained and the names of the variables AXL, AXR, AYU, AYD are retained even though these quantities are now conductivities rather than diffusivities. Conductivities are set to either that of frozen material or that of thawed material depending on whether the cell being computed or its neighbor is frozen or thawed. Since the conductivity of thawed material is substantially less than that of frozen material, the thawed material is assumed to control the flow of heat whenever one cell is thawed and its neighbor is frozen. This subroutine also sets the value of density times specific heat for the cell being computed.

All calculations related to melting and freezing are handled in subroutine MELFRZ. The new and old temperatures are tested to see if a melting or freezing process is continuing, or whether the onset of melting or freezing is indicated. If there is no phase change, the new temperature replaces the old, and control returns to the calling program.

If there is melting or freezing, the calories entered or lost from the cell are calculated and added to or subtracted from the internal energy total. Completion of melting is indicated when the internal energy count passes its upper limit. When this happens the internal energy is set to the upper limit, and any excess calories are applied to raising the temperature of the thawed material. Similarly, the completion of freezing is indicated when the internal energy count passes through zero. In this case, the internal energy of the cell is set to zero, and any excess calories lost are applied to lowering the temperature of the frozen material.

If an incremental addition or subtraction of calories is greater than one third of the calorie limit for the cell, a warning message is printed, and a count is kept of the number of such warnings. If the number of warnings exceeds 100, the job is aborted.

Initial ambient temperature as a function of time and depth is calculated in subroutine AMB and initial internal energies are calculated in subroutine ENERGY.

Subroutine CELLD is used to calculate cell dimensions when a variable grid is employed. Currently this subroutine is not used, and the calls to it have been embedded in comment delimiters. The program provides for three levels of grid sizes in both the x and y directions which may be used if the subroutine CELLD is reactivated. The calls to CELLD are located in subroutines FORMA, FORMB, FORMC, FORMD, FORME, FORMF, and ENERGY. In the x direction, the border between the first and second band of grids is indicated by the variable NLR1; the border between the second and third band is indicated by NLR2. In the y direction, the border between the first and second band is indicated by NLM, and the border between the second and third band is indicated by NLN. The grid interval in the second band is expected to be double that of the first, and the grid interval in the third band should be double that of the second.

Subroutine PRINT causes the printing of printer plots of isothermal lines. These plots give one grid interval resolution at best and are intended to provide for the pictorial monitoring of the temperature history with a minimum consumption of paper and printer time. These plots supplement the full numerical output of the temperature array and the internal-energy array that is made at each stop point in the history.

A compiled listing of the source program follows.

```
/* PIPELINE THERMAL ANALYSIS
.....................................................................

THIS PROGRAM COMPUTES THE TEMPERATURE HISTORY OF A CROSS-SECTION OF
PERMAFROST IN WHICH A PIPE IS BURIED AT A DEPTH OF 8 FEET.  THE PIPE
IS MAINTAINED AT A CONSTANT TEMPERATURE OF 80 DEGREES CENTIGRADE.
THE SURFACE OF THE PERMAFROST IS SUBJECT TO SEASONAL VARIATION IN
TEMPERATURE.  THE EQUATION OF HEAT CONDUCTION IN 2-DIMENSIONAL
CARTESIAN COORDINATES IS SOLVED BY THE ALTERNATING-DIRECTION EXPLICIT
METHOD TO OBTAIN THE TEMPERATURE HISTORY.  OUTPUT IS IN THE FORM OF
PRINTER PLOTS OF ISOTHERMAL LINES.
                                        PROGRAMMED BY P.C. DOHERTY

.....................................................................
                                                                    */
HOTPIPE:   PROCEDURE OPTIONS(MAIN);
           DECLARE (NCL,NROW,NCOL,NLM,NLN)            FIXED BIN (31);
           DECLARE (NLR1,NLR2)                        FIXED BIN (31);
           DECLARE (DT,HDT,DXA,DYA,DXB,DYB,DXC,DYC)   FLOAT DEC (16);
           DECLARE (NPT,NPB,KSTEP,NSTOP)              FIXED BIN (31);
           DECLARE  NTIME     FLOAT DEC (16);
           DECLARE (TIME,XX,YY,DXL,DXR,DYU,DYD)       FLOAT DEC (16);
           DECLARE (TMPIPE,PCT,HOM,PER,AMP,TMELT)     FLOAT DEC (16);
           DECLARE (DENF,DENM,CNDF,CNDM,SPHF,SPHM)    FLOAT DEC (16);
           DECLARE (PI,TPI,ALPF,ALPM)                 FLOAT DEC (16);
           DECLARE (AREA,XCAL,CLIM,TNEW,NUM,DOM)      FLOAT DEC (16);
           DECLARE (AXL,AXR,AYU,AYD)                  FLOAT DEC (16);
           DECLARE (KPX(135),KYEAR,KFRAC)             FIXED BIN (31);
           DECLARE (IMAGE(131),IMAGX(131),IMAGY(131)) CHAR(1);
           DECLARE  SYMB(20)  CHAR (1);
           DECLARE  SYMT(20)  FLOAT DEC (16);
           DECLARE (NSYM,KPRINT,NPRINT) FIXED BIN (31);
           DECLARE (LAMBDA,TMAMB)                     FLOAT DEC (16);
           DECLARE  KSTOP     FIXED BIN (31);
           DECLARE  YEARS                             FLOAT DEC (16);
           DECLARE (RPIPA,TPIOP)   FLOAT DEC (16);
           DECLARE (I,J,K,II,JJ,MM,NN,KSYM)           FIXED BIN (31);
           DECLARE (A,B,C)   FLOAT DEC (16);
           DECLARE (DELCAL,ERRORCNT,ERAT,ERATM)       FLOAT DEC (16);
           DECLARE (RCF,RCM,RC)                       FLOAT DEC (16);
           OPEN FILE(SYSPRINT) LINESIZE(131);
           PI=3.1415926535E0;
           TPI=2.0E0*PI;
           TMAMB=0.0E0;
           LAMBDA=0.0E0;
                      /*** READ BASIC PARAMETERS ***/
READIN:    ON ENDFILE (SYSIN) GOTO FINISH;
           GET FILE (SYSIN) DATA;
           KSTOP=NSTOP*NPRINT;
           NPT=NCL-1;
           NPB=NCL+1;
```

```
            DT=PER/(NTIME*NPRINT);
            HDT=DT/2.0E0;
            TPIOP=TPI/PER;
            ALPF=CNDF/(DENF*SPHF);
            ALPM=CNDM/(DENM*SPHM);
            RPIPA=SQRT(PI/(PER*ALPF));
            DYU=DYA;
            DYD=DYA;
            DXL=DXA;
            DXR=DXA;
            AREA=(DXL+DXR)*(DYU+DYD)/4.0E0;
            CLIM=DENF*AREA*PCT*HOM;
            RCF=DENF*SPHF;
            RCM=DENM*SPHM;
                        /*** SET UP PRINT-COLUMN INDEX ARRAY ***/
            DO I=1 TO NLR1;
            KPX(I)=I;
            END;
            IF  NLR2<=NLR1 THEN GOTO ENDKPX;
            DO I=NLR1+1 TO NLR2;
            KPX(I)=NLR1+2*(I-NLR1);
            END;
            IF  NCOL<=NLR2  THEN GOTO ENDKPX;
            DO I=NLR2+1 TO NCOL;
            KPX(I)=NLR1+2*(NLR2-NLR1)+4*(I-NLR2);
            END;
ENDKPX:     ;
                        /*** SET UP PLOT IMAGES ***/
            IMAGX=' ';
            IMAGY=' ';
            DO I=1 TO KPX(NCOL);
            IMAGX(I)='-';
            END;
            DO I=1 TO KPX(NCOL) BY 10;
            IMAGX(I)='|';
            IMAGY(I)='|';
            END;
                        /*** PRINT TITLE PAGE ***/
            PUT FILE(SYSPRINT) EDIT
                  ('X','X','XXXXXXXX','XXXXXXXX','XXXXXXXX',
                  'X','X','X','X','X',
                  'X','X','XXXXXXXX','X','XXXX','XXXXXXXX',
                  'X','X','X','X','X','X',
                  'XXXXXX','X','XXXXXXXX','X','XXXXXXXX','X',
                  'XXXXXXXX','X')
                  (PAGE,SKIP(6),X(30),A,X(6),A,X(12),A,X(12),A,X(12),A,
                  SKIP(1),X(30),A,X(6),A,X(12),A,X(19),A,X(19),A,
                  SKIP(1),X(30),A,X(6),A,X(12),A,X(12),A,X(3),A,X(12),A,
                  SKIP(1),X(30),A,X(6),A,X(19),A,X(12),A,X(6),A,X(19),A,
                  SKIP(1),X(31),A,X(3),A,X(9),A,X(2),A,X(9),A,X(2),A,
                  X(9),A,X(2),A)
```

```
        ('MENLO PARK, CALIFORNIA') (SKIP(3),X(53),A)
        (('*' DO I=1 TO 57)) (SKIP(8),X(36),57 A)
        ('*','*') (SKIP(1),X(36),A,X(55),A)
        ('*  THERMAL ANALYSIS OF PROPOSED TRANS-ALASKAN ',
        'PIPELINE  *') (SKIP(1),X(36),A,A)
        ('*','*') (SKIP(1),X(36),A,X(55),A)
        (('*' DO J=1 TO 57)) (SKIP(1),X(36),57 A);
              /*** PRINT PROBLEM PARAMETERS ***/
    PUT FILE(SYSPRINT) EDIT
        ('U. S.  GEOLOGICAL SURVEY  ---  MENLO PARK, CALIF',
        'ORNIA  ---  AUGUST, 1969')
        (PAGE,A,A)
        ('THERMAL ANALYSIS OF PROPOSED TRANS-ALASKAN ',
        'PIPELINE') (SKIP(2),A,A)
        ('PROBLEM PARAMETERS....') (SKIP(3),A);
              /*** PHYSICAL PARAMETERS ***/
    PUT FILE(SYSPRINT) EDIT
        ('PIPE TEMPERATURE (DEG C)=',TMPIPE)
        (SKIP(2),A,F(7,2))
        ('MELTING TEMPERATURE OF WATER (DEG C)=',TMELT)
        (SKIP(1),A,F(5,2))
        ('LATENT HEAT OF MELTING (CAL/GM)=',HOM)
        (SKIP(1),A,F(6,2))
        ('MOISTURE CONTENT OF GROUND (GM/GM)=',PCT)
        (SKIP(1),A,F(6,3))
        ('AVERAGE AMBIENT TEMPERATURE (DEG C)=',TMAMB)
        (SKIP(1),A,F(7,3))
        ('AMPLITUDE OF AMBIENT TEMPERATURE VARIATION ',
        '(DEG C)=',AMP) (SKIP(1),A,A,F(7,3))
        ('PERIOD OF AMBIENT TEMPERATURE VARIATION (SEC)=',
        PER) (SKIP(1),A,E(13,6))
        ('PHASE ANGLE OF AMBIENT TEMPERATURE VARIATION ',
        '(RADIANS)=',LAMBDA) (SKIP(1),A,A,F(8,5))
        ('DENSITY OF FROZEN GROUND (GM/CC)=',DENF)
        (SKIP(1),A,F(6,3))
        ('DENSITY OF THAWED GROUND (GM/CC)=',DENM)
        (SKIP(1),A,F(6,3))
        ('THERMAL CONDUCTIVITY OF FROZEN GROUND ',
        '(CAL/CM/SEC/DEG C)=',CNDF) (SKIP(1),A,A,F(7,4))
        ('THERMAL CONDUCTIVITY OF THAWED GROUND ',
        '(CAL/CM/SEC/DEG C)=',CNDM) (SKIP(1),A,A,F(7,4))
        ('SPECIFIC HEAT OF FROZEN GROUND (CAL/GM/DEG C)=',
        SPHF) (SKIP(1),A,F(6,3))
        ('SPECIFIC HEAT OF THAWED GROUND (CAL/GM/DEG C)=',
        SPHM) (SKIP(1),A,F(6,3))
        ('THERMAL DIFFUSIVITY OF FROZEN GROUND (CM*CM/SEC)=',
        ALPF) (SKIP(1),A,F(7,4))
        ('THERMAL DIFFUSIVITY OF THAWED GROUND (CM*CM/SEC)=',
        ALPM) (SKIP(1),A,F(7,4));
              /*** COMPUTATIONAL PARAMETERS ***/
    PUT FILE(SYSPRINT) EDIT
```

```
                ('GRID POSITION OF PIPE CENTER=',NCL)
                (SKIP(2),A,F(2))
                ('GRID ROWS=',NROW) (SKIP(1),A,F(2))
                ('GRID COLUMNS=',NCOL) (SKIP(1),A,F(2))
                ('HORIZONTAL GRID INTERVALS=',DXA,DXB,DXC)
                (SKIP(1),A,3 F(10,4))
                ('VERTICAL GRID INTERVALS=',DYA,DYB,DYC)
                (SKIP(1),A,3 F(10,4))
                ('INTEGRATION STEP SIZE (SEC)=',DT)
                (SKIP(1),A,E(13,6))
                ('STEPS PER PRINT=',NPRINT) (SKIP(1),A,F(2))
                ('PRINTS PER PERIOD=',NTIME) (SKIP(1),A,F(6,2))
                ('TOTAL NUMBER OF PRINTS=',NSTOP) (SKIP(1),A,F(3));
                    /*** SET UP VARIABLE ARRAYS ***/
(NOUNDERFLOW):
CONTROL:    BEGIN;
            DECLARE (T(NROW,NCOL),E(NROW,NCOL))        FLOAT DEC (16);
            ON OVERFLOW  GOTO BUGOUT;
                    /*** SET UP AMBIENT TEMPERATURES ***/
            TIME=0;
            KSTEP=0;
            KPRINT=0;
            CALL AMB;
            DO I=1 TO NCOL-1;
            T(*,I)=T(*,NCOL);
            END;
            T(NROW,*)=TMAMB;
                    /*** SET PIPE TEMPERATURES ***/
            DO J=NPT TO NPB;
            DO I=1 TO 2;
            T(J,I)=TMPIPE;
            END; END;
                    /*** SET INTERNAL ENERGIES ***/
            CALL ENERGY;
                    /*** PRINT ISOTHERMS ***/
            CALL PRINT;
                    /*** PRINT NUMBERS ***/
            PUT FILE(SYSPRINT) EDIT ('TIME=',TIME) (PAGE,A,E(13,6));
            DO I=1 TO NCOL;
            PUT FILE(SYSPRINT) EDIT ('T,COL=',I) (SKIP(2),A,F(3))
                ((T(J,I) DO J=1 TO NROW))
                (SKIP(1),10 F(13,6));
            END;
                    /*** ADVANCE TIME ***/
LOOP:       KSTEP=KSTEP+1;
            KPRINT=KPRINT+1;
            /*** ADVANCE SURFACE BOUNDARY ***/
            TIME=TIME+HDT;
            I=1;
            YY=0.0E0;
            CALL AMBTMP;
```

```
T(1,*)=T(1,NCOL);
                /*** DOWNWARD SWEEP ***/
CALL DNSWEEP;
/*** ADVANCE SURFACE BOUNDARY ***/
TIME=TIME+HDT;
I=1;
YY=0.0E0;
CALL AMBTMP;
T(1,*)=T(1,NCOL);
                /*** UPWARD SWEEP ***/
CALL UPSWEEP;
                /*** PRINT ISOTHERMS ***/
IF  KPRINT>=NPRINT  THEN DO;
KPRINT=0;
CALL PRINT;
END;
                /*** TEST FOR FINISH ***/
IF  KSTEP<KSTOP  THEN GOTO LOOP;
                /*** PRINT NUMBERS ***/
PUT FILE(SYSPRINT) EDIT ('TIME=',TIME) (PAGE,A,E(13,6));
DO I=1 TO NCOL;
PUT FILE(SYSPRINT) EDIT ('T,COL=',I) (SKIP(2),A,F(3))
      ((T(J,I) DO J=1 TO NROW))
      (SKIP(1),10 E(13,6));
END;
DO I=1 TO NCOL;
PUT FILE(SYSPRINT) EDIT ('F,COL=',I) (SKIP(2),A,F(3))
      ((F(J,I) DO J=1 TO NROW))
      (SKIP(1),10 E(13,6));
END;
                /*** READ NEW NTIME,NPRINT,NSTOP ***/
GET FILE(SYSIN) DATA;
                /*** TEST FOR END OF HISTORY ***/
IF  NTIME=0  THEN GOTO READIN;
                /*** SET NEW STEP SIZES ***/
DT=PER/(NTIME*NPRINT);
HDT=DT/2.0E0;
KSTOP=KSTOP+NSTOP*NPRINT;
PUT FILE(SYSPRINT) EDIT
      ('U. S.  GEOLOGICAL SURVEY  ---  MENLO PARK, CALIF',
      'ORNIA  ---  AUGUST, 1969')
      (PAGE,A,A)
      ('THERMAL ANALYSIS OF PROPOSED TRANS-ALASKAN ',
      'PIPELINE') (SKIP(2),A,A)
      ('PROBLEM PARAMETERS....') (SKIP(3),A);
PUT FILE(SYSPRINT) EDIT
      ('INTEGRATION STEP SIZE (SEC)=',DT)
      (SKIP(2),A,F(13,6))
      ('STEPS PER PRINT=',NPRINT) (SKIP(1),A,F(2))
      ('PRINTS PER PERIOD=',NTIME) (SKIP(1),A,F(6,2))
      ('TOTAL NUMBER OF PRINTS=',NSTOP) (SKIP(1),A,F(3));
```

```
            GOTO LOOP;
                        /*** DOWNWARD SWEEP OF ADE ***/
DNSWEEP:    PROCEDURE;
            DO J=2 TO NPT-1;
            I=1;
            CALL FORMA;
            DO I=2 TO NCOL-1;
            CALL FORMB;
            END;
            I=NCOL;
            CALL FORMC;
            END;
            DO J=NPT TO NPB;
            DO I=3 TO NCOL-1;
            CALL FORMB;
            END;
            I=NCOL;
            CALL FORMC;
            END;
            DO J=NPB+1 TO NROW-1;
            I=1;
            CALL FORMA;
            DO I=2 TO NCOL-1;
            CALL FORMB;
            END;
            I=NCOL;
            CALL FORMC;
            END;
            RETURN;
            END DNSWEEP;
                        /*** UPWARD SWEEP OF ADE ***/
UPSWEEP:    PROCEDURE;
            DO JJ=1 TO NROW-NPB-1;
            J=NROW-JJ;
            I=NCOL;
            CALL FORMD;
            DO II=1 TO NCOL-2;
            I=NCOL-II;
            CALL FORME;
            END;
            I=1;
            CALL FORMF;
            END;
            DO JJ=NROW-NPB TO NROW-NPT;
            J=NROW-JJ;
            I=NCOL;
            CALL FORMD;
            DO II=1 TO NCOL-3;
            I=NCOL-II;
            CALL FORME;
            END;  END;
```

```
                DO JJ=NROW-NPT+1 TO NROW-2;
                J=NROW-JJ;
                I=NCOL;
                CALL FORMD;
                DO II=1 TO NCOL-2;
                I=NCOL-II;
                CALL FORME;
                END;
                I=1;
                CALL FORMF;
                END;
                RETURN;
                END UPSWEEP;
                        /*** TEMPERATURE AT CENTERLINE ***/
FORMA:          PROCEDURE;
                        /*** FIND CELL DIMENSIONS ***/
                /*  CALL CELLD  */
                        /*** FIND ALPHA ***/
                CALL ALPHA;
                        /*** COMPUTE NUMERATOR ***/
                NUM= RC*T(J,I)/HDT + AXR*(T(J,I+1)-T(J,I))/DXR**2
                   + AYD*(T(J+1,I)-T(J,I))/DYD**2 + AYU*T(J-1,I)/DYU**2;
                        /*** COMPUTE DENOMINATOR ***/
                DOM= RC/HDT     + AYU/DYU**2;
                        /*** COMPUTE NEW TEMPERATURE ***/
                TNEW= NUM/DOM;
                        /*** TEST FOR MELTING OR FREEZING ***/
                CALL MELFR7;
                RETURN;
                END FORMA;
                        /*** INTERIOR DOWNWARD TEMPERATURE ***/
FORMB:          PROCEDURE;
                        /*** FIND CELL DIMENSIONS ***/
                /*  CALL CELLD  */
                        /*** FIND ALPHA ***/
                CALL ALPHA;
                        /*** COMPUTE NUMERATOR ***/
                NUM= RC*T(J,I)/HDT + AXR*(T(J,I+1)-T(J,I))/DXR**2
                   + AXL*T(J,I-1)/DXL**2 + AYD*(T(J+1,I)-T(J,I))/DYD**2
                   + AYU*T(J-1,I)/DYU**2;
                        /*** COMPUTE DENOMINATOR ***/
                DOM= RC/HDT     + AXL/DXL**2 + AYU/DYU**2;
                        /*** COMPUTE NEW TEMPERATURE ***/
                TNEW= NUM/DOM;
                        /*** TEST FOR MELTING OR FREEZING ***/
                CALL MELFRZ;
                RETURN;
                END FORMB;
                        /*** DOWNWARD EASTERN BOUNDARY ***/
FORMC:          PROCEDURE;
                /*  CALL CELLD  */
```

```
                CALL ALPHA;
                        /*** COMPUTE NUMERATOR ***/
                NUM= RC*T(J,I)/HDT + AXL*T(J,I-1)/DXL**2
                   + AYD*(T(J+1,I)-T(J,I))/DYD**2 + AYU*T(J-1,I)/DYU**2;
                        /*** COMPUTE DENOMINATOR ***/
                DOM= RC/HDT     + AXL/DXL**2 + AYU/DYU**2;
                        /*** NEW TEMPERATURE ***/
                TNEW=NUM/DOM;
                        /*** TEST FOR MELTING OR FREEZING ***/
                CALL MELFRZ;
                RETURN;
                END FORMC;
                        /*** UPWARD EASTERN BOUNDARY ***/
FORMD:          PROCEDURE;
                /*  CALL CELLD  */
                CALL ALPHA;
                        /*** COMPUTE NUMERATOR ***/
                NUM= RC*T(J,I)/HDT + AXL*(T(J,I-1)-T(J,I))/DXL**2
                   + AYD*T(J+1,I)/DYD**2 + AYU*(T(J-1,I)-T(J,I))/DYU**2;
                        /*** COMPUTE DENOMINATOR ***/
                DOM= RC/HDT     + AYD/DYD**2;
                        /*** NEW TEMPERATURE ***/
                TNEW= NUM/DOM;
                        /*** TEST FOR MELTING OR FREEZING ***/
                CALL MELFRZ;
                RETURN;
                END FORMD;
                        /*** INTERIOR UPWARD TEMPERATURE ***/
FORME:          PROCEDURE;
                        /*** FIND CELL DIMENSIONS ***/
                /*  CALL CELLD  */
                        /*** FIND ALPHA ***/
                CALL ALPHA;
                        /*** COMPUTE NUMERATOR ***/
                NUM= RC*T(J,I)/HDT + AXR*T(J,I+1)/DXR**2
                   + AXL*(T(J,I-1)-T(J,I))/DXL**2 + AYD*T(J+1,I)/DYD**2
                   + AYU*(T(J-1,I)-T(J,I))/DYU**2;
                        /*** COMPUTE DENOMINATOR ***/
                DOM= RC/HDT     + AXR/DXR**2 + AYD/DYD**2;
                        /*** COMPUTE NEW TEMPERATURE ***/
                TNEW= NUM/DOM;
                        /*** TEST FOR MELTING OR FREEZING ***/
                CALL MELFRZ;
                RETURN;
                END FORME;
                        /*** TEMPERATURE AT CENTERLINE ***/
FORMF:          PROCEDURE;
                        /*** FIND CELL DIMENSIONS ***/
                /*  CALL CELLD  */
                        /*** FIND ALPHA ***/
                CALL ALPHA;
```

```
                        /*** COMPUTE NUMERATOR ***/
                NUM= RC*T(J,I)/HDT + AXR*T(J,I+1)/DXR**2
                   + AYD*T(J+1,I)/DYD**2 + AYU*(T(J-1,I)-T(J,I))/DYU**2;
                        /*** COMPUTE DENOMINATOR ***/
                DOM= RC/HDT     + AXR/DXR**2 + AYD/DYD**2;
                        /*** COMPUTE NEW TEMPERATURE ***/
                TNEW= NUM/DOM;
                        /*** TEST FOR MELTING OR FREEZING ***/
                CALL MELFRZ;
                RETURN;
                END FORMF;
                        /*** FIND ALPHA VALUES ***/
ALPHA:          PROCEDURE;
                        /*** LEFT-HAND ALPHA VALUE ***/
                IF  I=1  THEN DO;
                IF  T(J,I)<TMELT  THEN AXL=CNDF;
                ELSE AXL=CNDM;
                END;
                ELSE DO;
                IF  T(J,I)>TMELT  |  T(J,I-1)>TMELT  THEN AXL=CNDM;
                ELSE AXL=CNDF;
                END;
                        /*** RIGHT-HAND ALPHA VALUE ***/
                IF  I=NCOL  THEN DO;
                IF  T(J,I)<TMELT  THEN AXR=CNDF;
                ELSE AXR=CNDM;
                END;
                ELSE DO;
                IF  T(J,I)>TMELT  |  T(J,I+1)>TMELT  THEN AXR=CNDM;
                ELSE AXR=CNDF;
                END;
                        /*** UPPER ALPHA VALUE ***/
                IF  J=1  THEN DO;
                IF  T(J,I)<TMELT  THEN AYU=CNDF;
                ELSE AYU=CNDM;
                END;
                ELSE DO;
                IF  T(J,I)>TMELT  |  T(J-1,I)>TMELT  THEN AYU=CNDM;
                ELSE AYU=CNDF;
                END;
                        /*** LOWER ALPHA VALUE ***/
                IF  J=NROW  THEN DO;
                IF  T(J,I)<TMELT  THEN AYD=CNDF;
                ELSE AYD=CNDM;
                END;
                ELSE DO;
                IF  T(J,I)>TMELT  |  T(J+1,I)>TMELT  THEN AYD=CNDM;
                ELSE AYD=CNDF;
                END;
                        /*** SET DEN*SPH ***/
                IF  T(J,I)>TMELT  THEN RC=RCM;
```

```
                ELSE RC=RCF;
                RETURN;
                END ALPHA;
                        /*** MELTING-FREEZING MANAGEMENT ***/
MELFRZ:         PROCEDURE;
                        /*** TEST FOR CONTINUING MELT OR FREEZE ***/
                IF  T(J,I)=TMELT  THEN GOTO CONTIN;
                        /*** TEST FOR ONSET OF MELTING ***/
                IF  T(J,I)<TMELT  &  TNEW>TMELT  THEN DO;
                T(J,I)=TMELT;
                GOTO CONTIN;
                END;
                        /*** TEST FOR ONSET OF FREEZING ***/
                IF  T(J,I)>TMELT  &  TNEW<TMELT  THEN DO;
                T(J,I)=TMELT;
                GOTO CONTIN;
                END;
                        /*** NO PHASE CHANGE ***/
                T(J,I)=TNEW;
                RETURN;
                        /*** TEST FOR FREEZE OR MELT ***/
CONTIN:         IF  TNEW>TMELT  THEN DO;
                        /*** MELT ***/
                        /*** CALORIE CHANGE ***/
                DELCAL=DENF*AREA*SPHF*(TNEW-TMELT);
                        /*** TEST FOR EXCESSIVE CALORIE CHANGE ***/
                ERAT=ABS(DELCAL)/CLIM;
                IF  ERAT>ERATM  THEN ERATM=ERAT;
                IF  ERAT>0.333E0  THEN DO;
                PUT FILE(SYSPRINT) EDIT ('CALORIE INCREMENT TOO BIG',
                'ROW=',J,'COL=',I,'T=',T(J,I),'E=',E(J,I),'LIMIT=',CLIM,
                'CALORIE INCREMENT=',DELCAL)
                (SKIP(2),A,X(2),A,F(3),X(2),A,F(3),SKIP(1),A,E(13,6),X(2),
                A,E(13,6),X(2),A,E(13,6),X(2),A,E(13,6));
                ERRORCNT=ERRORCNT+1.0E0;
                IF  ERRORCNT>1.0E2  THEN EXIT;
                END;
                        /*** ADD TO CALORIE COUNT ***/
                E(J,I)=E(J,I)+DELCAL;
                        /*** TEST FOR COMPLETION OF MELT ***/
                IF  E(J,I)>=CLIM  THEN DO;
                XCAL=E(J,I)-CLIM;
                E(J,I)=CLIM;
                T(J,I)=TMELT+XCAL/(DENF*AREA*SPHM);
                END;
                RETURN;
                END;
                ELSE DO;
                        /*** FREEZE ***/
                        /*** CALORIE CHANGE ***/
                DELCAL=DENF*AREA*SPHF*(TNEW-TMELT);
```

```
                        /*** TEST FOR EXCESSIVE CALORIE CHANGE ***/
                ERAT=ABS(DELCAL)/CLIM;
                IF  ERAT>ERATM  THEN ERATM=ERAT;
                IF  ERAT>0.333E0  THEN DO;
                PUT FILE(SYSPRINT) EDIT ('CALORIE INCREMENT TOO BIG',
                'ROW=',J,'COL=',I,'T=',T(J,I),'E=',E(J,I),'LIMIT=',CLIM,
                'CALORIE INCREMENT=',DELCAL)
                (SKIP(2),A,X(2),A,F(3),X(2),A,F(3),SKIP(1),A,E(13,6),X(2),
                A,E(13,6),X(2),A,F(13,6),X(2),A,E(13,6));
                ERRORCNT=ERRORCNT+1.0E0;
                IF  ERRORCNT>1.0E2   THEN EXIT;
                END;
                        /*** ADD TO CALORIE COUNT ***/
                E(J,I)=E(J,I)+DELCAL;
                        /*** TEST FOR COMPLETION OF FREEZE ***/
                IF  E(J,I)<=0.0E0  THEN DO;
                XCAL=E(J,I);
                E(J,I)=0.0E0;
                T(J,I)=TMELT+XCAL/(DENF*AREA*SPHF);
                END;
                RETURN;
                END;
                END MELFRZ;
                        /*** INITIALIZE INTERNAL ENERGY ***/
ENERGY:         PROCEDURE;
                E=0;
                DO J=1 TO NROW;
                DO I=1 TO NCOL;
                        /*** TEST AGAINST MELT TEMPERATURE ***/
                IF  T(J,I)>TMELT  THEN DO;
                /*  CALL CELLD  */
                E(J,I)=AREA*DENF*PCT*HOM;
                END;
                END;  END;
                RETURN;
                END ENERGY;
                        /*** CALCULATE CELL DIMENSIONS ***/
CELLD:          PROCEDURE;
                IF  J<NLM  THEN DO;
                DYU=DYA;
                DYD=DYA;
                GOTO NEXT;
                END;
                IF  J=NLM  THEN DO;
                DYU=DYA;
                DYD=DYB;
                GOTO NEXT;
                END;
                IF  J<NLN  THEN DO;
                DYU=DYB;
                DYD=DYB;
```

```
                GOTO NEXT;
                END;
                IF  J=NLN  THEN DO;
                DYU=DYB;
                DYD=DYC;
                GOTO NEXT;
                END;
                DYU=DYC;
                DYD=DYC;
NEXT:           ;
                IF  I<NLR1  THEN DO;
                DXL=DXA;
                DXR=DXA;
                GOTO FIN;
                END;
                IF  I=NLR1  THEN DO;
                DXL=DXA;
                DXR=DXB;
                GOTO FIN;
                END;
                IF  I<NLR2  THEN DO;
                DXL=DXB;
                DXR=DXB;
                GOTO FIN;
                END;
                IF  I=NLR2  THEN DO;
                DXL=DXB;
                DXR=DXC;
                GOTO FIN;
                END;
                DXL=DXC;
                DXR=DXC;
FIN:            ;
                AREA=(DXL+DXR)*(DYU+DYD)/4.0E0;
                CLIM=DENF*AREA*PCT*HOM;
                RETURN;
                END CELLD;
                        /*** AMBIENT TEMPERATURES ***/
AMB:            PROCEDURE;
                DO I=1 TO NLM;
                YY=(I-1)*DYA;
                CALL AMBTMP;
                END;
                IF  NLN<=NLM  THEN GOTO AMBEND;
                DO I=NLM+1 TO NLN;
                YY=(NLM-1)*DYA+(I-NLM)*DYB;
                CALL AMBTMP;
                END;
                IF  NROW<=NLN  THEN GOTO AMBEND;
                DO I=NLN+1 TO NROW;
                YY=(NLM-1)*DYA+(NLN-NLM)*DYB+(I-NLN)*DYC;
```

```
                CALL AMBTMP;
                END;
AMBEND:         ;
                RETURN;
                END AMB;
                        /*** SEASONAL TEMPERATURE ***/
AMBTMP:         PROCEDURE;
                IF  AMP=0.0E0  THEN DO;
                T(I,NCOL)=TMAMB;
                RETURN;
                END;
                A=SIN(TPIOP*TIME+LAMBDA-YY*RPIPA);
                B=EXP(-YY*RPIPA);
                T(I,NCOL)=AMP*A*B+TMAMB;
                RETURN;
                END AMBTMP;
                        /*** PRINT ISOTHERMS ***/
PRINT:          PROCEDURE;
                        /*** FIND YEARS AND FRACTIONS ***/
                YEARS=TIME/PER;
                KYEAR=YEARS+0.01E0;
                KFRAC=YEARS*NTIME-KYEAR*NTIME+0.01E0;
                        /*** PRINT TITLE ***/
                PUT FILE(SYSPRINT) EDIT ('TIME=',KYEAR,'AND',KFRAC,
                      '/',NTIME,'YEARS','OR',YEARS,'YEARS',
                      'MAX F-RATIO=',ERATM)
                      (PAGE,A,F(3),X(1),A,X(1),F(2),A,F(4,1),X(1),A,X(2),
                      A,X(1),F(7,3),X(1),A,X(3),A,E(13,6))
                      ((SYMB(I),'=',SYMT(I),'DEGREES' DO I=1 TO NSYM))
                      (SKIP(1),7 (A,A,F(5,1),X(1),A,X(3)));
                PUT FILE(SYSPRINT) EDIT (' ') (SKIP(1),A);
                        /*** BEGIN PLOT ***/
                J=0;
LOOPA:          J=J+1;
                IMAGE=IMAGX;
                K=1;
                GOTO INSERT;
LOOPB:          J=J+1;
                K=K+1;
                IMAGE=IMAGY;
INSERT:         ;
                        /*** TEST FOR LINE SKIP ***/
                IF  J>NLN  THEN DO KSKP=1 TO 3;
                PUT FILE(SYSPRINT) EDIT ((IMAGE(I) DO I=1 TO 131))
                      (SKIP(1),13] A);
                IF  K<10  THEN DO;
                K=K+1;
                IMAGE=IMAGY;
                END;
                ELSE DO;
                K=1;
```

```
            IMAGE=IMAGX;
            END;
            END;
            ELSE IF  J>NLM  THEN DO;
            PUT FILE(SYSPRINT) EDIT ((IMAGE(I) DO I=1 TO 131))
            (SKIP(1),131 A);
            IF  K<10  THEN DO;
            K=K+1;
            IMAGE=IMAGY;
            END;
            ELSE DO;
            K=1;
            IMAGE=IMAGX;
            END;
            END;
                        /*** MARK POSITIVE TEMPERATURES ***/
            DO I=1 TO NCOL;
            IF  T(J,I)>0.0E0  THEN IMAGE(KPX(I))='+';
            END;
                        /*** INSERT SYMBOLS ***/
            DO KSYM=1 TO NSYM;
                        /*** TEMPERATURE SEARCH ***/
            I=1;
            IF  T(J,I)=SYMT(KSYM)  THEN DO;
            IMAGE(KPX(I))=SYMB(KSYM);
            GOTO REST;
            END;
            IF  J>1  THEN DO;
            IF  T(J-1,I)<SYMT(KSYM) & T(J,I)>SYMT(KSYM) THEN DO;
            IMAGE(KPX(I))=SYMB(KSYM);
            GOTO REST;
            END;
            IF  T(J-1,I)>SYMT(KSYM) & T(J,I)<SYMT(KSYM) THEN DO;
            IMAGE(KPX(I))=SYMB(KSYM);
            GOTO REST;
            END;  END;
REST:       DO I=2 TO NCOL;
            IF  T(J,I)=SYMT(KSYM)  THEN DO;
            IMAGE(KPX(I))=SYMB(KSYM);
            GOTO ENDSYM;
            END;
            IF  T(J,I-1)<SYMT(KSYM) & T(J,I)>SYMT(KSYM) THEN DO;
            IMAGE(KPX(I))=SYMB(KSYM);
            GOTO ENDSYM;
            END;
            IF  T(J,I-1)>SYMT(KSYM) & T(J,I)<SYMT(KSYM) THEN DO;
            IMAGE(KPX(I))=SYMB(KSYM);
            GOTO ENDSYM;
            END;
ENDSYM:     END;
            END;
```

```
                    /*** PRINT LINE ***/
              PUT FILE(SYSPRINT) EDIT ((IMAGE(I) DO I=1 TO 131))
                    (SKIP(1),131 A);
                    /*** TEST AND LOOP ***/
              IF  J<NROW  THEN DO;
              IF  K<10  THEN GOTO LOOPB;
              GOTO LOOPA;
              END;
              FRATM=0.0E0;
              RETURN;
              END PRINT;
BUGOUT:       ;
              PUT FILE(SYSPRINT) EDIT ('OVERFLOW SENSED') (PAGE,A);
              PUT FILE(SYSPRINT) EDIT
                    ('KSTEP=',KSTEP) (SKIP(1),A,F(5))
                    ('I=',I,'J=',J) (SKIP(1),A,F(3),X(2),A,F(3))
                    ('NUM=',NUM,'DOM=',DOM,'TNEW',TNEW)
                    (SKIP(1),A,E(13,6),X(2),A,E(13,6),X(2),A,E(13,6))
                    ('T(J,I)=',T(J,I),'E(J,I)=',E(J,I))
                    (SKIP(1),A,E(13,6),X(2),A,E(13,6));
              DO I=1 TO NCOL;
              PUT FILE(SYSPRINT) EDIT ('T,COL=',I) (SKIP(2),A,F(3))
                    ((T(J,I) DO J=1 TO NROW))
                    (SKIP(1),10 E(13,6));
              END;
              DO I=1 TO NCOL;
              PUT FILE(SYSPRINT) EDIT ('E,COL=',I) (SKIP(2),A,F(3))
                    ((E(J,I) DO J=1 TO NROW))
                    (SKIP(1),10 E(13,6));
              END;
              END CONTROL;
FINISH:       RETURN;
              END HOTPIPE;
```

RESTRICTIONS

1. Grid dimensions are limited by memory space available. A 30x45 grid has been used in a region of 250K but no attempt has been made to find the largest grid that would fit in this region.

2. The user must be careful to insure that the right boundary and the bottom boundary are far enough distant from the pipe so as not to cause a serious error as the liquid-solid interface moves out. Distances of sixty feet for the right boundary and ninety feet for the bottom boundary have been found satisfactory for the problems studied to date.

3. The user must be careful to insure that the integration time step is not so large as to produce unacceptable truncation errors. Warning messages of rapid changes in internal energy may indicate serious truncation errors. Experience to date indicates that steps of 1/104 year are satisfactory for the first year of history, 1/52 year steps are satisfactory from one year to five years, and 1/26 year steps are satisfactory after five years.

INPUT REQUIREMENTS AND DATA DESCRIPTION

Input to this program is in data-directed, stream-input form. Input identifiers are punched on the input cards together with the input values. Items are separated by commas, and groups of items are separated by semicolons. Within a group, items may appear in any order and may be punched anywhere on the input data cards. Each item is terminated by a comma except the last item in a group which is terminated by a semicolon.

A set of input data for a 20-year history of interaction between the pipe and a typical interior silt is listed below:

Item	Remarks	Units
NCL=5,	Row index of pipe center.	
NROW=46,	Number of rows in grid.	
NCOL=31,	Number of columns in grid.	
NLM=46,	Boundary of small row grid.	
NLN=46,	Boundary of medium row grid.	
NLR1=31,	Boundary of small column grid.	
NLR2=31,	Boundary of medium column grid.	
DXA=60.96012E0,	Small column grid interval.	(cm)
DYA=60.96012E0,	Small row grid interval.	(cm)
DXB=121.92024E0,	Medium column grid interval.	(cm)
DYB=121.92024E0,	Medium row grid interval.	(cm)
DXC=243.84048E0,	Large column grid interval.	(cm)
DYC=243.84048E0,	Large row grid interval.	(cm)

Since NLM=NLN=NROW=46 and NLR1=NLR2=NCOL=31, only DXA and DYA are effective and we have a uniform square grid.

PER=3.154E7,	Period of seasonal temperature variation.	(sec)
AMP=20.0E0,	Amplitude of seasonal temperature variation.	(deg C)
LAMBDA=0.0E0,	Phase of seasonal temperature variation.	(rad)
TMPIPE=80.0E0,	Pipe temperature.	(deg C)
TMELT=0.0E0,	Melting temperature of water.	(deg C)
HOM=80.0E0,	Latent heat of melting for water.	(cal/gm)
TMAMB=-0.8E0,	Mean ambient temperature of ground.	(deg C)
PCT=0.15E0,	Moisture content of ground.	(gm/gm)
CNDF=0.00050E0,	Conductivity of frozen ground.	(cal/cm/sec/deg C)
CNDM=0.0034E0,	Conductivity of thawed ground.	(cal/cm/sec/deg C)
DENF=1.9E0,	Density of frozen ground.	(gm/cc)
DENM=1.9E0,	Density of thawed ground.	(gm/cc)
SPHF=0.22E0,	Specific heat of frozen ground.	(cal/gm/deg C)
SPHM=0.30E0,	Specific heat of thawed ground.	(cal/gm/deg C)
NSYM=5,	Number of isotherms to be plotted.	
SYMB(1)='#',	Symbol for first isotherm.	
SYMT(1)=+5.0E0,	Temperature of first isotherm.	(deg C)
SYMB(2)='X',	Symbol for second isotherm.	
SYMT(2)=-0.5E0,	Temperature of second isotherm.	(deg C)
SYMB(3)='@',	Symbol for third isotherm.	
SYMT(3)=-0.8E0,	Temperature of third isotherm.	(deg C)
SYMB(4)='0',	Symbol for fourth isotherm.	
SYMT(4)=0.0E0,	Temperature of fourth isotherm.	(deg C)
SYMB(5)='*',	Symbol for fifth isotherm.	
SYMT(5)=80.0E0,	Temperature of fifth isotherm.	(deg C)
NTIME=1.0E0,	Number of prints per period.	
NPRINT=104,	Number of steps per print.	
NSTOP=1;	Number of prints to stop.	

This is the end of the first group of data items. The integration will advance over a one-year time span at 1/104 year steps, printing a plot at the end of each year, after which the temperature array and energy array are printed in full.

```
NTIME=1.0E0,
NPRINT=52,
NSTOP=4;
```
This carries the integration up to five years.

```
NTIME=1.0E0,
NPRINT=26,
NSTOP=5;
```
This continues the history up to ten years.

```
NTIME=12.0E0,
NPRINT=4,
NSTOP=1;
```
Twelve copies of this card cause monthly outputs of plots and arrays during the eleventh year of history.

```
NTIME=1.0E0,
NPRINT=26,
NSTOP=9;
```
This continues the history up to twenty years.

```
NTIME=12.0E0,
NPRINT=4,
NSTOP=1;
```
Twelve copies of this card cause monthly outputs of plots and arrays during the twenty-first year of history.

```
NTIME=0;
```
Signals the end of the history.

PROGRAM RUN PREPARATION

The deck setup for compile and execute is as follows:

```
//   Standard JOB card
//STEP1 EXEC PROC=PL1LFCLG,REGION.GO=250K,TIME.GO=35
//PL1L.SYSIN DD *

       PL/1 source program

/*
//GO.SYSIN DD *

       Input data cards

/*
```

All input is from cards; all output is on the printer.

OUTPUT

Printed output includes a title page, problem-identifying parameters, a plot of initial conditions, the initial temperature array, plots and array prints as prescribed by input parameters. Samples of the output follow this page. In the printer plots, thawed cells are identified by the symbol "+", unless overridden by some other symbol.

The parameter MAX E-RATIO is printed with each plot. This is the maximum ratio of energy increment to energy limit that has been calculated since the previous plot output. This parameter gives the user some feeling for the number of integration steps required to melt a cell.

MENLO PARK, CALIFORNIA

****

*

* THERMAL ANALYSIS OF PROPOSED TRANS-ALASKAN PIPELINE *

*

****

U. S. GEOLOGICAL SURVEY --- MENLO PARK, CALIFORNIA --- AUGUST, 1969

THERMAL ANALYSIS OF PROPOSED TRANS-ALASKAN PIPELINE

PROBLEM PARAMETERS....

PIPE TEMPERATURE (DEG C)= 80.00
MELTING TEMPERATURE OF WATER (DEG C)= 0.00
LATENT HEAT OF MELTING (CAL/GM)= 80.00
MOISTURE CONTENT OF GROUND (GM/GM)= 0.150
AVERAGE AMBIENT TEMPERATURE (DEG C)= -8.900
AMPLITUDE OF AMBIENT TEMPERATURE VARIATION (DEG C)= 20.000
PERIOD OF AMBIENT TEMPERATURE VARIATION (SEC)= 3.154000E+07
PHASE ANGLE OF AMBIENT TEMPERATURE VARIATION (RADIANS)= 0.00000
DENSITY OF FROZEN GROUND (GM/CC)= 1.900
DENSITY OF THAWED GROUND (GM/CC)= 1.900
THERMAL CONDUCTIVITY OF FROZEN GROUND (CAL/CM/SEC/DEG C)= 0.0050
THERMAL CONDUCTIVITY OF THAWED GROUND (CAL/CM/SEC/DEG C)= 0.0034
SPECIFIC HEAT OF FROZEN GROUND (CAL/GM/DEG C)= 0.220
SPECIFIC HEAT OF THAWED GROUND (CAL/GM/DEG C)= 0.300
THERMAL DIFFUSIVITY OF FROZEN GROUND (CM*CM/SEC)= 0.0120
THERMAL DIFFUSIVITY OF THAWED GROUND (CM*CM/SEC)= 0.0060

GRID POSITION OF PIPE CENTER= 5
GRID ROWS=25
GRID COLUMNS=25
HORIZONTAL GRID INTERVALS= 60.9601 121.9202 243.8405
VERTICAL GRID INTERVALS= 60.9601 121.9202 243.8405
INTEGRATION STEP SIZE (SEC)= 3.032692E+05
STEPS PER PRINT=04
PRINTS PER PERIOD= 1.00
TOTAL NUMBER OF PRINTS= 1

TIME= 0 AND 0/ 1.0 YEARS OR 0.000 YEARS MAX F-RATIO= 0.000000E+00
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES d= -8.9 DEGREES

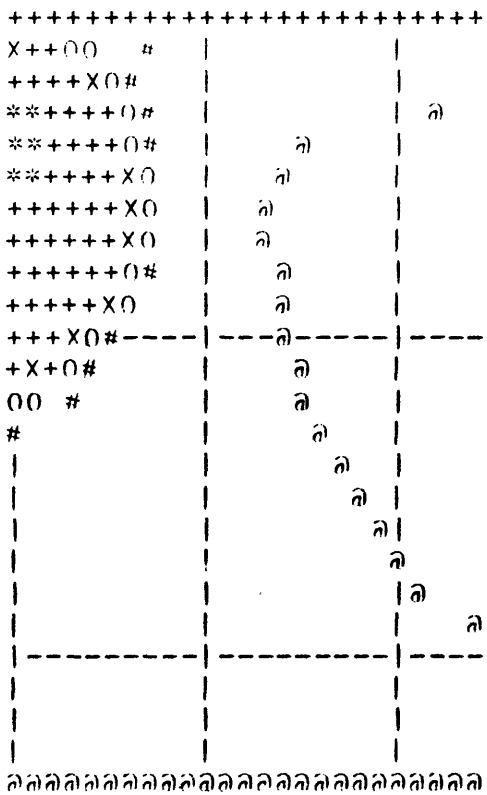
U. S. GEOLOGICAL SURVEY --- MENLO PARK, CALIFORNIA --- AUGUST, 1969

THERMAL ANALYSIS OF PROPOSED TRANS-ALASKAN PIPELINE

PROBLEM PARAMETERS....

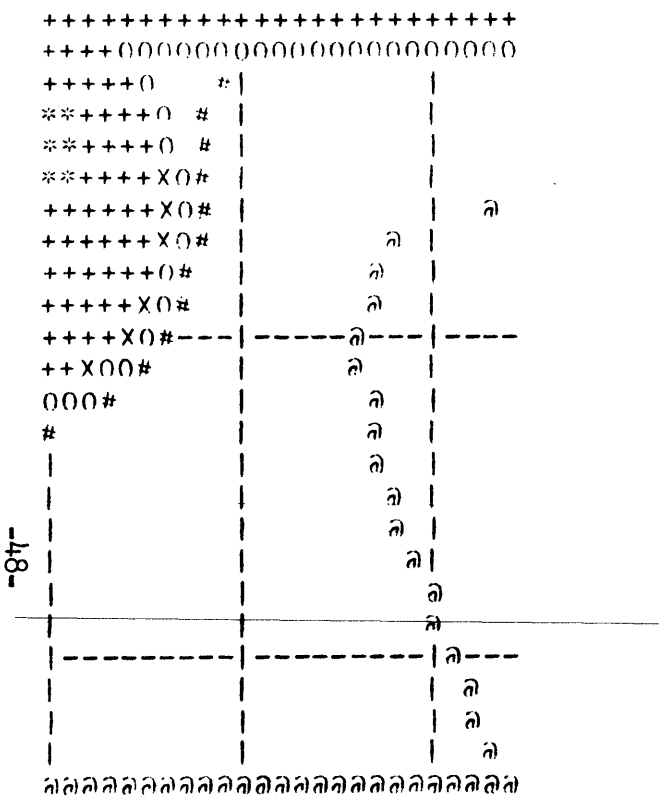
INTEGRATION STEP SIZE (SEC)= 6.065385E+05
STEPS PER PRINT= 4
PRINTS PER PERIOD= 13.00
TOTAL NUMBER OF PRINTS= 13

TIME= 1 AND 1/13.0 YEARS OR 1.077 YEARS MAX F-RATIO= 1.598502E-01
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES

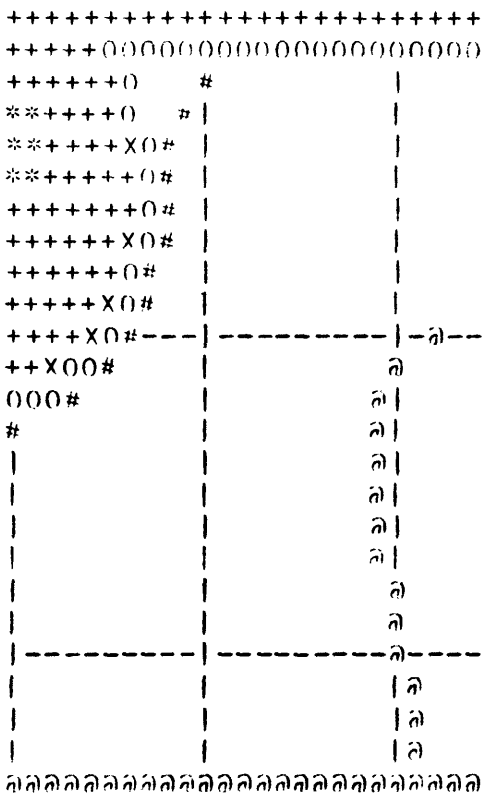


TIME= 1 AND 2713.0 YEARS OR 1.154 YEARS MAX F-RATIO= 1.918784E-01

X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES

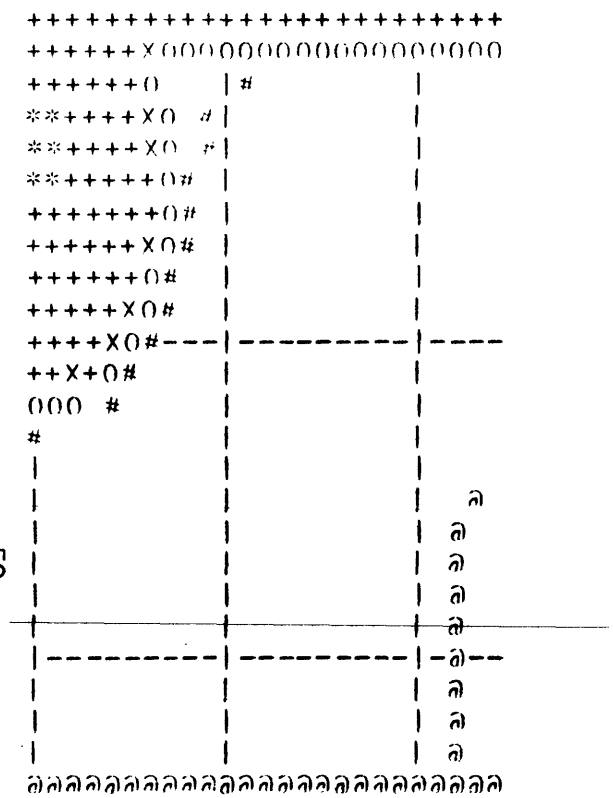


TIME= 1 AND 3/13.0 YEARS OR 1.231 YEARS MAX E-RATIO= 1.671991E-01
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES

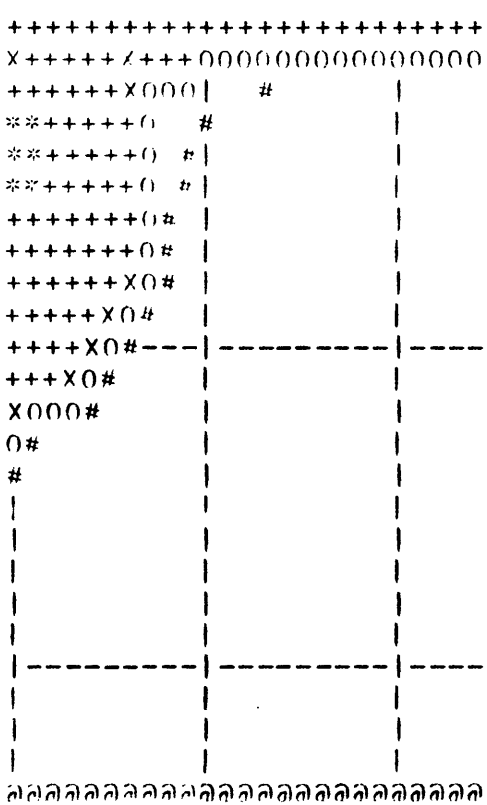


TIME= 1 AND 4/13.0 YEARS OR 1.308 YEARS MAX E-RATIO= 1.449626E-01

X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES

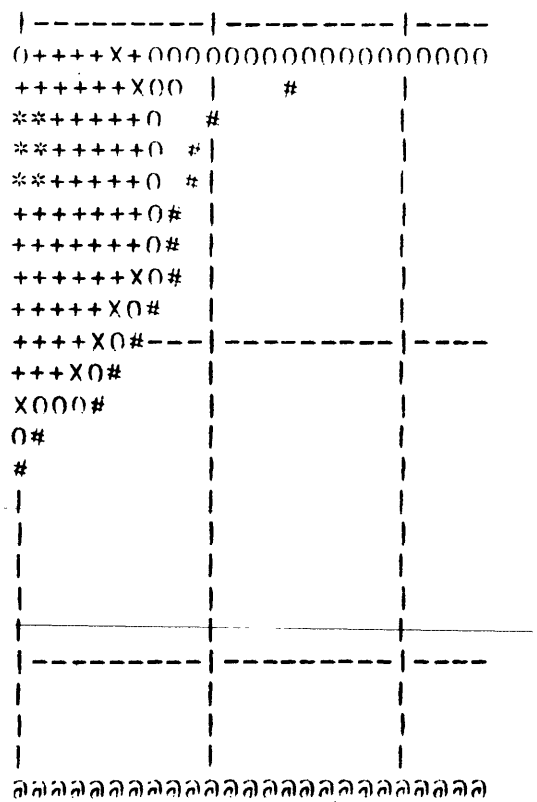


TIME= 1 AND 5/13.0 YEARS OR 1.385 YEARS MAX E-RATIO= 1.028944E-01
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES m= -8.9 DEGREES

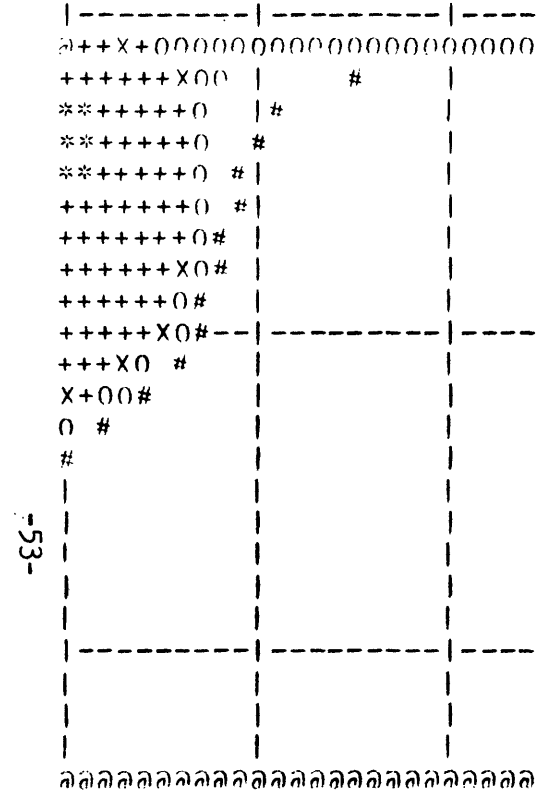


TIME= 1 AND 6/13.0 YEARS OR 1.462 YEARS MAX E-RATIO= 4.469017E-02

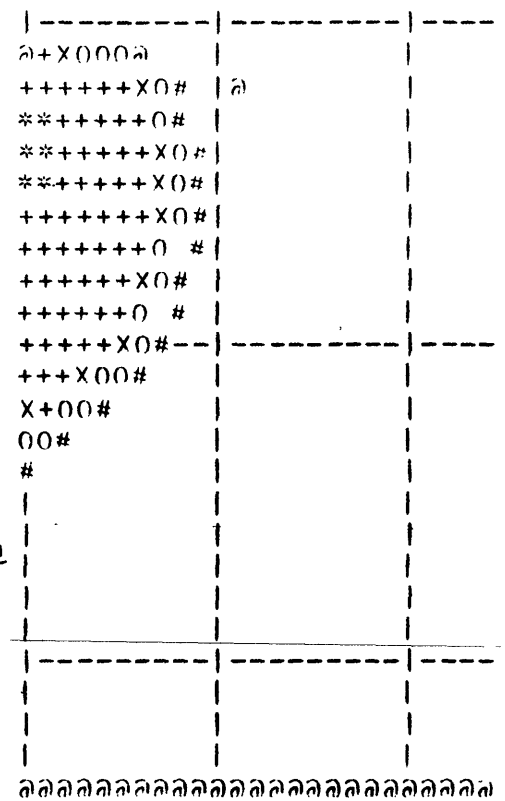
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES



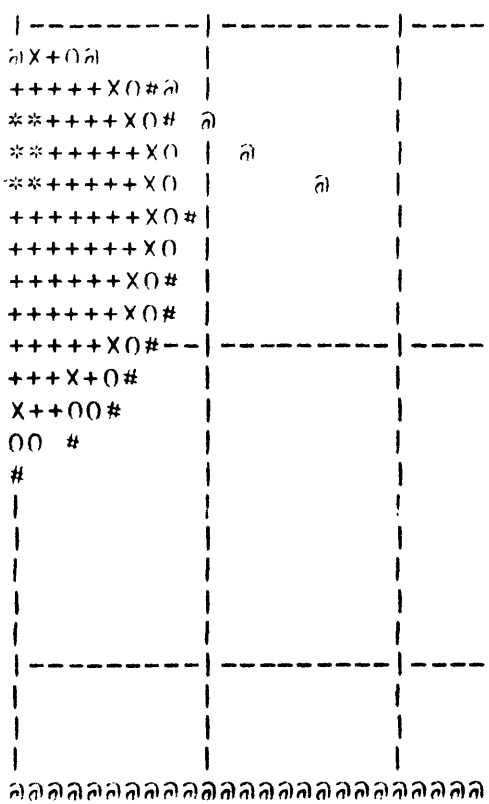
TIME= 1 AND 7/13.0 YEARS OR 1.538 YEARS MAX E-RATIO= 1.3045091E-01
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES



TIME= 1 AND 8/13.0 YEARS OR 1.615 YEARS MAX E-RATIO= 1.914451E-01
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES

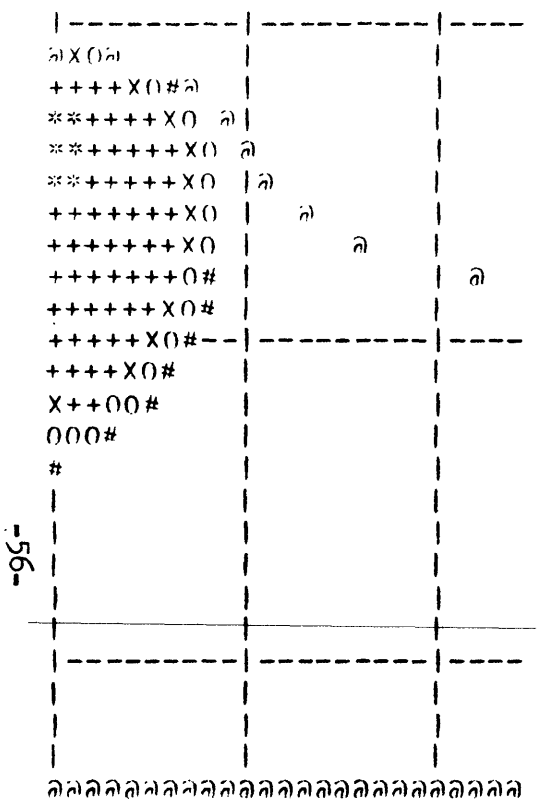


TIME= 1 AND 9/13.0 YEARS OR 1.692 YEARS MAX E-RATIO= 1.975711E-01
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES

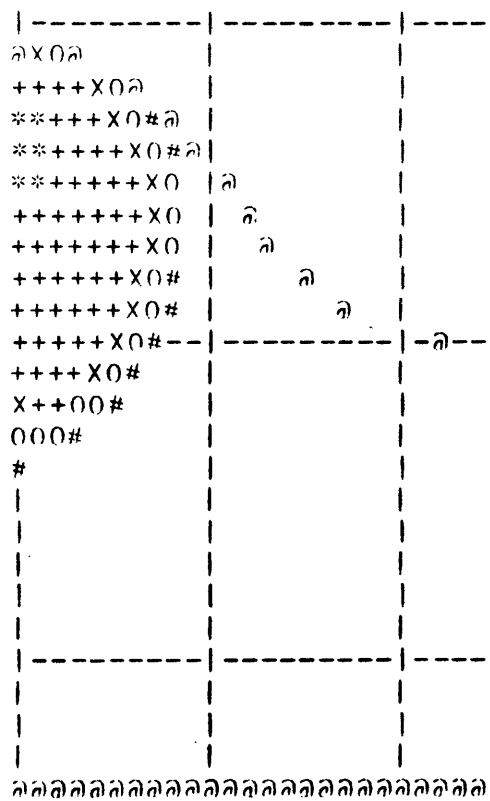


TIME= 1 AND 10/13.0 YEARS OR 1.769 YEARS MAX F-RATIO= 1.384109E-01

X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES

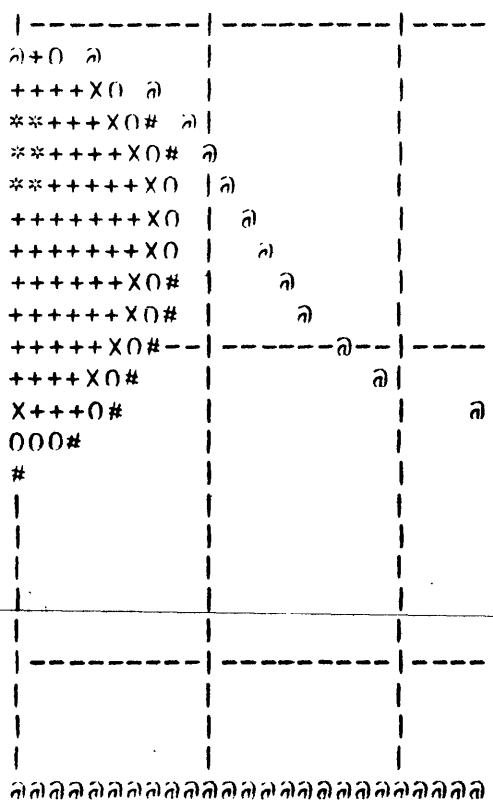


TIME= 1 AND 11/13.0 YEARS OR 1.846 YEARS MAX E-RATIO= 1.361431E-01
X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES



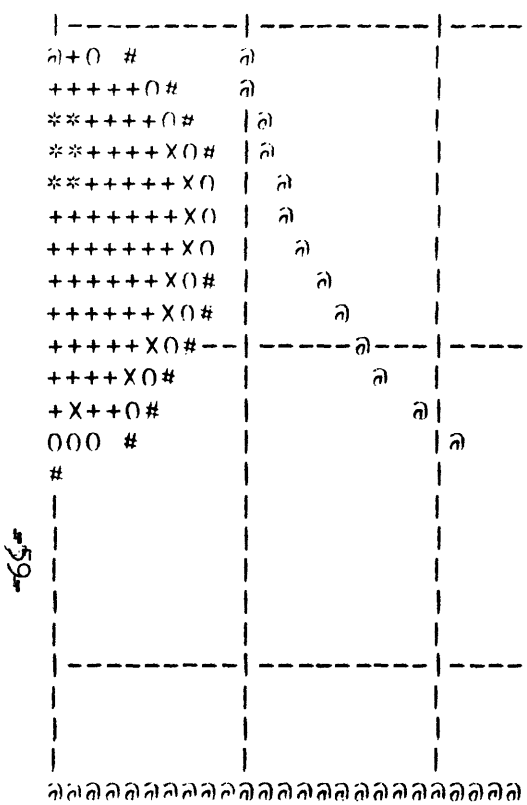
TIME= 1 AND 12/13.0 YEARS OR 1.923 YEARS MAX E-RATIO= 5.704024E-02

X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES



TIME= 2 AND 0/13.0 YEARS OR 2.000 YEARS MAX F-RATIO= 1.067508E-01

X= 5.0 DEGREES #= -0.5 DEGREES 0= 0.0 DEGREES *= 80.0 DEGREES @= -8.9 DEGREES



```
TIME= 6.308000E+07

T,COL=  1
-8.900000E+00 1.564661E+01 4.503111E+01 8.000000E+01 8.000000E+01 8.00000
 1.704656E+01 1.068000E+01 5.125104E+00 0.000000E+00-1.016147E+00-1.99508
-6.121225E+00-6.837200E+00-7.534694E+00-8.220258E+00-8.900000E+00

T,COL=  2
-8.900000E+00 1.157861E+01 4.032180E+01 8.000000E+01 8.000000E+01 8.00000
 1.588268E+01 9.913706E+00 4.725727E+00 0.000000E+00-1.053304E+00-2.04755
-6.145607E+00-6.854793E+00-7.546068E+00-8.225826E+00-8.900000E+00

T,COL=  3
-8.900000E+00 0.000000E+00 2.525758E+01 4.647579E+01 5.348715E+01 5.33526
 1.368076E+01 8.414739E+00 3.897325E+00 0.000000E+00-1.149559E+00-2.16694
-6.193833E+00-6.889516E+00-7.568500E+00-8.236811E+00-8.900000E+00

T,COL=  4
-8.900000E+00-1.042513E-01 1.434923E+01 2.742370E+01 3.428000E+01 3.56619
 1.068666E+01 6.219898E+00 2.483586E+00-1.570093E-01-1.378935E+00-2.38046
-6.264942E+00-6.940563E+00-7.601436E+00-8.252935E+00-8.900000E+00

T,COL=  5
-8.900000E+00-2.461599E+00 6.347827E+00 1.512559E+01 2.085268E+01 2.29945
 7.182988E+00 3.343350E+00 0.000000E+00-8.843291E-01-1.827370E+00-2.70704
-6.357127E+00-7.006587E+00-7.643991E+00-8.273764E+00-8.900000E+00

T,COL=  6
-8.900000E+00-4.320254E+00 0.000000E+00 6.505205E+00 1.133211E+01 1.36602
 3.495874E+00 0.000000E+00-7.744809E-01-1.552631E+00-2.339188E+00-3.10067
-6.467649E+00-7.085681E+00-7.694958E+00-8.298712E+00-8.900000E+00

T,COL=  7
-8.900000E+00-5.791752E+00-3.032177E+00 0.000000E+00 4.490323E+00 6.69334
 0.000000E+00-8.303665E-01-1.539703E+00-2.214347E+00-2.879382E+00-3.53283
-6.593062E+00-7.175495E+00-7.752866E+00-8.327074E+00-8.900000E+00

T,COL=  8
-8.900000E+00-6.870941E+00-5.069373E+00-2.577522E+00 0.000000E+00 1.50234
-1.136937E+00-1.773680E+00-2.343378E+00-2.891877E+00-3.438728E+00-3.98784
-6.729571E+00-7.273423E+00-7.816084E+00-8.358063E+00-8.900000E+00

T,COL=  9
-8.900000E+00-7.770914E+00-6.837401E+00-5.071674E+00-3.433164E+00-2.64974
-2.409650E+00-2.786350E+00-3.175894E+00-3.581850E+00-4.008028E+00-4.45398
-6.873334E+00-7.376791E+00-7.882922E+00-8.390861E+00-8.900000E+00

T,COL= 10
-8.900000E+00-8.515605E+00-8.296160E+00-7.128561E+00-5.951150E+00-5.04614
-3.639620E+00-3.790580E+00-4.003649E+00-4.267147E+00-4.574589E+00-4.92013
-7.020670E+00-7.483000E+00-7.951723E+00-8.424664E+00-8.900000E+00

T,COL= 11
-8.900000E+00-9.111381E+00-9.449074E+00-8.727989E+00-7.834444E+00-6.97412
-4.772841E+00-4.740033E+00-4.796935E+00-4.928925E+00-5.124998E+00-5.37568
-7.168194E+00-7.589633E+00-8.020937E+00-8.458714E+00-8.900000E+00

T,COL= 12
-8.900000E+00-9.577897E+00-1.034507E+01-9.964546E+00-9.289912E+00-8.51292
-5.789957E+00-5.610596E+00-5.535632E+00-5.551990E+00-5.647651E+00-5.81141
```

```
-8.311072E+00-8.427795E+00-8.570839E+00-8.731133E+00-8.900000E+00

T,COL =  25
-8.90  000E+00-1.109636E+01-1.326629E+01-1.406209E+01-1.426984E+01-1.40585
-1.073922E+01-1.010485E+01-9.558239E+00-9.108297E+00-8.756538E+00-8.49922
-8.321967E+00-8.435928E+00-8.576251E+00-8.733845E+00-8.900000E+00

F,COL =  1
 0.000000E+00 8.472791E+04 8.472791E+04 8.472791E+04 8.472791E+04 8.47279
 8.472791E+04 8.472791E+04 8.472791E+04 7.360372E+04 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  2
 0.000000E+00 8.472791E+04 8.472791E+04 8.472791E+04 8.472791E+04 8.47279
 8.472791E+04 8.472791E+04 8.472791E+04 3.498257E+04 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  3
 0.000000E+00 3.437493E+04 8.472791E+04 8.472791E+04 8.472791E+04 8.47279
 8.472791E+04 8.472791E+04 8.472791E+04 6.589593E+03 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  4
 0.000000E+00 0.000000E+00 8.472791E+04 8.472791E+04 8.472791E+04 8.47279
 8.472791E+04 8.472791E+04 8.472791E+04 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  5
 0.000000E+00 0.000000E+00 8.472791E+04 8.472791E+04 8.472791E+04 8.47279
 8.472791E+04 8.472791E+04 1.379044E+04 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  6
 0.000000E+00 0.000000E+00 7.710107E+01 8.472791E+04 8.472791E+04 8.47279
 8.472791E+04 6.388025E+04 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  7
 0.000000E+00 0.000000E+00 0.000000E+00 2.232537E+04 8.472791E+04 8.47279
 5.858391E+04 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  8
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 8.584955E+03 8.47279
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  9
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

F,COL =  10
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00
```



```
F,COL =  11
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00000
```

PROGRAM ERROR MESSAGES

1. "CALORIE INCREMENT TOO BIG"

   This is a warning message that the calorie change in some cell exceeds 1/3 the limiting value. A reduction in time step size is indicated.

2. "OVERFLOW SENSED"

   A floating-point overflow has occurred indicating some serious trouble. Diagnostic output following this message should help in finding the cause of the error.

TIMING

To compile and execute a 20-year history with substantial printer output required 35 minutes of 360/65 central processor time.

STORAGE REQUIREMENTS

Storage requirements for all parts of the program except the dynamically allocated temperature and energy arrays are listed on the following page.

All floating-point numbers are FLOAT DEC (16), the maximum precision available.

REFERENCE

Quon, D., Dranchuk, P. M., Allada, S. R., and Leung, P. K., 1965, A stable, explicit, computationally efficient method for solving two-dimensional mathematical models of petroleum reservoirs, Journal of Canadian Petroleum Technology, v. 4, no. 2, p. 530-535.

STORAGE REQUIREMENTS.

THE STORAGE AREA FOR THE PROCEDURE LABELLED HOTPIPE IS 2164 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 31 IS 144 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE BEGIN BLOCK AT STATEMENT NO. 78 IS 420 BYTES LONG.

THE STORAGE AREA FOR THE ON UNIT AT STATEMENT NO. 80 IS 152 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED DNSWEEP IS 224 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED UPSWEEP IS 224 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED FORMA IS 196 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED FORMB IS 196 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED FORMC IS 196 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED FORMD IS 196 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED FORME IS 196 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED FORMF IS 196 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED ALPHA IS 196 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED MELFRZ IS 212 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED ENERGY IS 236 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED CELLD IS 152 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED AMB IS 260 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED AMBTMP IS 212 BYTES LONG.

THE STORAGE AREA (IN STATIC) FOR THE PROCEDURE LABELLED PRINT IS 280 BYTES LONG.

THE PROGRAM CSECT IS NAMED HOTPIPE AND IS 26814 BYTES LONG.

THE STATIC CSECT IS NAMED HOTPIPE1 AND IS 2028 BYTES LONG.