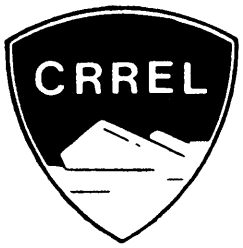
SR77-38

Special Report 77-38



FINITE ELEMENT MODEL OF TRANSIENT HEAT CONDUCTION WITH ISOTHERMAL PHASE CHANGE

(Two and Three Dimensional)

G.L. Guymon and T.V. Hromadka II

November 1977

CORPS OF ENGINEERS, U.S. ARMY
COLD REGIONS RESEARCH AND ENGINEERING LABORATORY
HANOVER, NEW HAMPSHIRE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Special Report 77-38	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* FINITE ELEMENT MODEL OF TRANSIENT HEAT CONDUCTION WITH ISOTHERMAL PHASE CHANGE (TWO AND THREE DIMENSIONAL)		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* G.L. Guymon and T.V. Hromadka II		8. CONTRACT OR GRANT NUMBER*(s)*
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California, Irvine Irvine, California		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Project 4A762719AT42 Task A3, Work Unit 006
11. CONTROLLING OFFICE NAME AND ADDRESS Directorate of Military Construction Office, Chief of Engineers Washington, D.C. 20314		12. REPORT DATE November 1977 13. NUMBER OF PAGES 167
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)* U.S. Army Cold Regions Research and Engineering Laboratory Hanover, New Hampshire 03755		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)*

Frost
Mathematical models
Soils
Thermal conductivity

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

The partial differential equation for transient heat conduction is solved by a finite element analog using a quadratic weighting function for the discretized spatial domain. The transient problem is solved by the Crank-Nicolson approximation. Two-dimensional and three-dimensional models incorporated in the same computer program are presented. The finite element method is reviewed, assumptions and limitations upon which the model is based are presented, and a complete derivation of the system analog is included. Certain problems can

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. Abstract (cont'd)

only be modeled as a three-dimensional system, e.g. thaw degradation around roadway culverts, embankment dams on permafrost where dam length is short relative to dam width, and thaw and freezeback under buildings. In most cases, however, the more economical two-dimensional model can be used. Numerical tests of both models have been accomplished but field verification has not been attempted. A user's manual and a FORTRAN IV computer listing of the program are presented.

PREFACE

This report was prepared by Dr. G.L. Guymon and T.V. Hromadka II of the School of Engineering, University of California, Irvine.

This report covers work funded by DA Project 4A762719AT42, Design, Construction and Operations Technology for Cold Regions, Task A3, Facilities Technology/Cold Regions, Work Unit 006, Volume Change Induced by Freezing and Thawing of Pavement Systems. The research consisted of two components: 1) the initial stages of development of a two-dimensional and three-dimensional heat transport model (with phase change) for freezing soils, and 2) the continued development of a one-dimensional frost heave model based upon solution of the coupled heat transport and fluid transport problem. Only the first element of research is reported herein. The second element of the research is being reported separately in the form of sections of another report.

It is emphasized that this report covers the initial phases of the development of a multidimensional heat transport model. While the techniques used herein are valid, it is expected that subsequent versions of the model will be prepared to increase efficiency and accuracy. In particular, more accurate techniques of handling the phase change problem can be envisioned and these techniques will be likely to reduce computer storage requirements and require less computer time. Subsequent improvements of the model will be reported upon in appropriate CRREL reports.

Cameron Appel of CRREL technically reviewed the manuscript of this report.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

This report presents a model of transient heat conduction in a freezing and thawing soil. The partial differential equation for transient heat conduction is solved by a finite element analog using a quadratic weighting function for the discretized spatial domain. The transient problem is solved by the Crank-Nicolson approximation. Phase change is approximated as an isothermal process.

Both a two-dimensional and three-dimensional model, incorporated in the same computer program, are presented. In the latter case, it is anticipated that certain problems can only realistically be modeled as a three-dimensional system. Examples of such problems include: thaw degradation around roadway culverts, embankment dams on permafrost where the dam length is short relative to the dam width, and thaw or freezeback under buildings. However, in many cases the more economical two-dimensional model may be used. Examples of such problems include: embankment dams of great length, roadway cross-sections, and long pipeline problems.

This report develops the basic equation of heat transport and the assumptions and limitations upon which the model is based. The finite element method is reviewed and a complete derivation of the system analog is presented. Numerous model evaluations were made and are summarized herein. It is emphasized that the primary thrust of evaluations is the numerical testing of the model. Field verification has not been attempted yet. A user manual and

computer listing of the program, written in standard FORTRAN IV, are presented. Although the model was developed on an IBM 370/155 computer, there should be little difficulty in adapting the model to other computers such as the Dartmouth System.

The main advantage of the model presented here is that it can be readily adapted to complex shapes which sometimes is a problem with the finite difference methods. The model can accommodate variable element sizes and configurations using triangular shaped and/or rectangular shaped elements in the two-dimensional case and tetrahedra shaped and/or brick shaped elements in the three-dimensional case.

The program has been prepared in a highly efficient manner, minimizing as much as possible computer execution time and minimizing the storage required for arrays.

Future work on the model will require field testing and verification. Additionally, it is desirable to couple a more sophisticated boundary conditional routine to this model in order to more readily simulate the soil air interface. The present model only handles a specified boundary condition or a no heat flux boundary condition.

Chapter 2

HEAT CONDUCTION EQUATION

A rigorous derivation of the heat transport equation can be found in Bird *et al.* (1960). For purposes of this report a more simplistic, but correct, derivation will be presented. Generally, most references, for example, Myers (1971), begin by making the deterministic-continuum assumption which usually leads to a partial differential equation with temperature as the state variable and various state parameters, e.g., heat capacity, that arise out of necessary mathematical-physical assumptions.

The first concept that needs to be employed is that energy is conserved. Thus, by considering the various rate processes involved in a particular process and by making an energy balance on a control volume, the appropriate heat equation is obtained. The various rate processes that might be considered are: conduction, convection, radiation, heat storage, and heat generation (e.g. latent heat effects).

The primary processes in a given soil system include all of these processes if the soil is freezing or thawing. Moreover, the soil system includes a heterogeneous mixture of dissimilar materials: mineral soil, organic material, air, water, and ice. Moreover, the water is often a dilute solution containing dissolved minerals which affect the system's thermal properties. The thermodynamics of soil systems is treated by Edelfsen and Anderson (1943) among others.

The derivation below will ignore radiation since this process occurs at

the soil surface. Radiation will be included in the system boundary conditions. Momentarily we will ignore energy generation in the soil due to freezing or thawing. Thus, consider a three-dimensional elemental volume of material in the presence of a fluid flux field. That is, fluid is moving through the elemental volume.

The energy balance equation is

$$\dot{E}_c + \dot{E}_v = \dot{E}_t \qquad (1)$$

where $\dot{E}_c$ is the net rate of heat conduction into the elemental volume, $\dot{E}_v$ is the net rate of heat convection into the elemental volume, and $\dot{E}_t$ is the total rate heat energy is stored in the elemental volume. Consider the x-direction

$$\left(\frac{q|_{x+\Delta x} - q|_x}{\Delta x} - \frac{[c_w v_x (T - T_o)|_{x+\Delta x} - c_w v_x (T - T_o)|_x]}{\Delta x} \right) \Delta y \Delta z \Delta x$$

$$= \text{net heat conduction and convection for the x-direction}$$

where $\Delta x \Delta y \Delta z$ is the volume of the element, T_o is a reference temperature, c_w is the volumetric heat capacity of the fluid and v_x is the fluid flux in the x-direction. The variable q may be replaced with Fourier's Law; i.e.,

$$q_x = -k_x \frac{\partial T}{\partial x}$$

where k is the thermal conductivity of the entire mass of material in the elemental volume. Substituting this law into the above yields

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) - \frac{\partial}{\partial x}\ (c_w v_x T)$$

for the rate of conduction and convection in the x-direction. Now considering the y- and x-directions in turn, similar expressions are added together and equated to the rate of heat energy accumulations in the elemental volume. Thus,

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right) - \frac{\partial}{\partial x}\left(c_w v_x T\right)$$

$$- \frac{\partial}{\partial y}\left(c_w v_y T\right) - \frac{\partial}{\partial z}\left(c_w v_z T\right) = c_a \frac{\partial T}{\partial \theta}$$

where c_a is the volumetric heat capacity of the mixture and θ is time.

Now looking at the convection term and expanding by the chain rule assuming an incompressible fluid (i.e. c_w = constant)

$$c_w\left(\frac{\partial\ (v_x T)}{\partial x} + \frac{\partial\ (v_y T)}{\partial y} + \frac{\partial\ (v_z T)}{\partial z}\right) =$$

$$c_w\left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z}\right) + c_w T\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right)$$

However, the last term is equal to zero since it is none other than the continuity expression for an incompressible fluid. Thus, the final result is

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right)$$

$$- c_w \left(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z}\right) = c_a \frac{\partial T}{\partial \theta} \qquad (2)$$

A major assumption that will be incorporated in the model developed herein is that the model will be developed for a system in which fluid flow is negligible. Therefore, the convective terms will be ignored and the following equation will be solved in general:

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right) = c_a \frac{\partial T}{\partial \theta} \qquad (3)$$

This equation will be solved in both three- and two-dimensions. The two-dimensional form is derived from Equation 3 by assuming $\partial T/\partial Z = 0$ and that in the z-direction the system consists of a slab of uniform thickness. Thus,

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) = c_a \frac{\partial T}{\partial \theta} \qquad (4)$$

Both equations are of the parabolic type. Boundary conditions to be considered are specified conditions and no heat conduction conditions. Initial conditions

are also required. Latent heat generation will be approximated as an isothermal process as will be described subsequently.

Chapter 3

FINITE ELEMENT METHOD

Introduction

The finite-element method has been routinely used in the structural, mechanical, and aerospace engineering fields since the use of modern digital computers became widespread. The finite-element literature relative to these fields is extensive. Following the mid-1960's, investigators concerned with general field problems became interested in this powerful numerical tool, and the volume of published articles dealing with the numerical solution of general problems by the finite-element method has increased markedly. Several recent texts present an excellent treatment of the finite-element method: Meyers (1971), Desai and Abel (1972), Zienkiewicz et al. (1971), Huebner (1975), and Sergerlind (1976). The latter two texts are especially recommended for the beginner.

The finite-element method is ideally suited to deal with complex geometries, anisotropy, and heterogeneity which are characteristic of most practical problems. For certain classes of problems, the finite-element method may be a more efficient numerical technique than the traditional finite-difference methods. That is, the finite-element computer program may require less execution time than the finite-difference computer program for a specified level of precision. In particular, time-dependent problems are efficiently solved by the finite-element method. The finite-difference methods often present stability and truncation error difficulties. Finite-element methods can

readily use arbitrary mesh spacing and can easily handle complicated boundary conditions that usually require lengthy programming by finite-difference methods. The finite-element program, once written for a class of problems, is very effective since it can be used to solve similar problems for any geometry and any mesh configuration desired. In other words, the finite-element program is completely general for the class of problems it was designed for. Finite-difference programs are often special purpose since they apply only to a specific geometry with a specific mesh spacing. For simple geometries and steady-state conditions, the finite-element methods appear to offer little advantage over the more familiar relaxation and iteration methods commonly applied to Laplace-like problems. Unfortunately, research on the various numerical methods has not advanced to the point where definitive criteria can be stated for the selection of the best method to use in a particular case. Indeed, there is still a good deal of controversy over the comparative advantages and disadvantages of the various methods.

There are two general approaches to the finite-element method: (i) the direct approach which involves writing a set of system matrices by visualizing the physical linkages of a system and (ii) the variational approach which involves developing a variational principle or applying the Galerkin technique. The direct approach is primarily applicable to structural engineering. The second approach consists of formulating a variational principle or formulating a governing partial differential equation which can be converted to an equivalent variational problem. It is the latter method, a mathematical abstraction, that is applicable to general field problems such as heat

transport.

For the problem considered in this report, the Galerkin method and the variational principle method lead to identical results. The variational principle method will be used to solve the problem. An extremum problem replaces the given partial differential equation, and a functional is found such that the extremum function also satisfies the given differential equation and its auxiliary conditions. That is, given the following functional in two-dimensional Cartesian coordinates

$$\iint F\left(x,y,T,\frac{\partial T}{\partial x},\frac{\partial T}{\partial y}\right)dxdy; \qquad \text{in R}$$

find a function T in R such that the functional is a minimum. If such a function exists, application of the fundamental lemma of integral calculus yields the so-called Euler equation; i.e.,

$$\frac{\partial}{\partial x}\left(F_{\partial T/\partial x}\right) + \frac{\partial}{\partial y}\left(F_{\partial T/\partial y}\right) - \frac{\partial F}{\partial T} = 0$$

It is this equation that is given to start with for field problems. In order to develop the finite-element method, we resort to a mathematical abstraction. We convert the governing partial-differential equation describing our problem to an entirely new problem, a variational problem.

In general, the finite-element method will be applied to the linear two-dimensional parabolic partial-differential equation for nonsteady heat

transport in an incompressible porous medium. For generality, a source term will be included and the medium will be considered anisotropic and heterogeneous. The method can readily be specialized to the one-dimensional case or steady-state case and can readily be extended to the more general three-dimensional case. The Cartesian coordinate system will be used, assuming the reader can extend the derivations here to other orthogonal coordinate systems. In essence, a specific problem is chosen as an example around which to develop the salient aspects of the finite-element technique.

A Variational Principle

Many problems of heat flow in porous materials are represented by the partial-differential equation

$$\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + Q = \rho c \frac{\partial T}{\partial \theta}$$

where T is a continuous, single-valued function (the unknown state variable); k_x and k_y are the thermal conductivities in the x and y directions, respectively; c is the heat capacity; Q is a generalized source term; and θ is time. An equivalent variational functional is

$$\chi = \iint \left[\frac{k_x}{2}\left(\frac{\partial T}{\partial x}\right)^2 + \frac{k_y}{2}\left(\frac{\partial T}{\partial y}\right)^2 + \left(\rho c \frac{\partial T}{\partial \theta} - Q\right) T \right] dx dy$$

where $\partial T/\partial\theta$ is assumed invariant or replaced by a finite difference analog, such as the Crank-Nicolson scheme. However, $\partial T/\partial\theta$ will be considered as invariant and the time-domain problem will be dealt with later.

Before continuing with the variational procedure, the problem must be more carefully specified. This consists of considering typical boundary conditions, and interface conditions. Interface conditions are required by the variational technique that is to be used for mathematical convenience. It is much too restrictive to apply the variational functional to the entire domain of interest, since it would be required that the first-order space partials exist throughout the domain. It is more convenient, and incidentally more practical for field problems, to consider it as applying to particular subregions of the domain, where subregions are separated by an interface, for example, abrupt changes in material properties.

Consider the connected domain R shown in Fig. 1, with boundaries and interfaces as shown. In each subregion, R_m, the partial-differential equation holds and on exterior boundary surfaces the following boundary conditions hold:

$$\partial T/\partial n = 0 \qquad \text{on } \Gamma_n, t \geqslant 0$$

$$T = h_s(S) \qquad \text{on } \Gamma_s, t \geqslant 0$$

The first condition is called a natural boundary condition and the second condition is called a geometrical boundary condition. Interface conditions

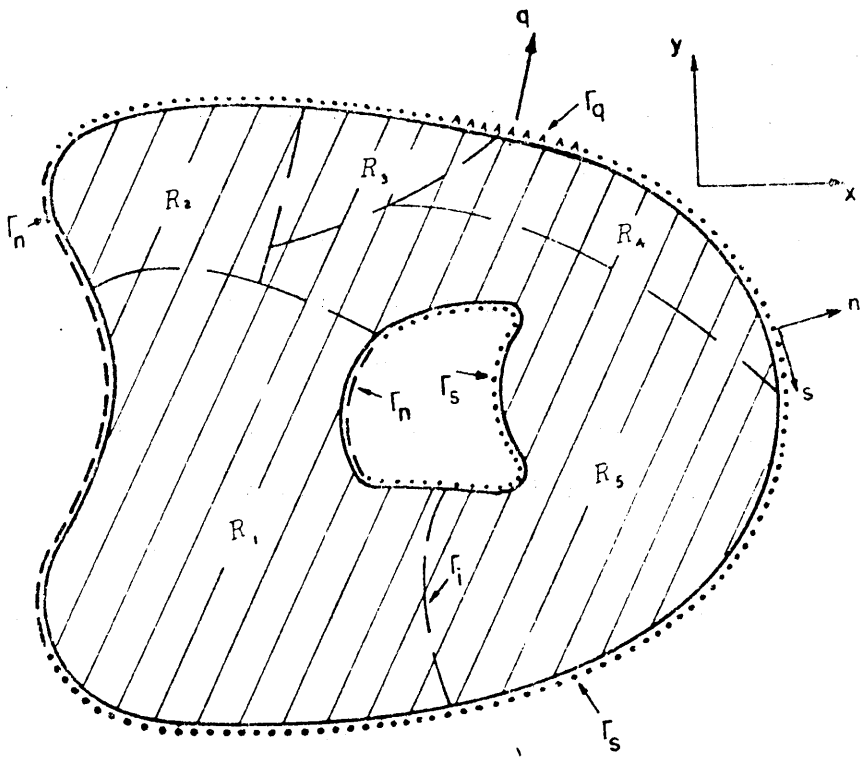


Fig. 1. Domain R

on Γ_i are

$$T\big|_{n^+} = T\big|_{n^-}, \qquad t \geqslant 0$$

and

$$k_n \frac{\partial T}{\partial n}\bigg|_{n^+} = k_n \frac{\partial T}{\partial n}\bigg|_{n^-}, \qquad t \geqslant 0$$

where n is the normal direction to the surface and s is the direction along the surface. Additionally, initial conditions are required throughout R i.e.

$$T\big|_{\theta=0} = T_0(x,y)$$

The above auxiliary conditions are applicable to a wide variety of situations encountered in the field. These conditions are chosen as an example and are not the only auxiliary conditions that can be dealt with by the finite-element method.

It is more convenient to rewrite the variational priniciple in the following equivalent form:

$$\chi = \sum_{m=1}^{M} \iint_{R_m} F_m \left(x,y,T, \frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}\right) dxdy$$

where

$$F_m\left(x,y,T,\frac{\partial T}{\partial x},\frac{\partial T}{\partial y}\right) = \frac{k_x}{2}\left(\frac{\partial T}{\partial x}\right)^2 + \frac{k_y}{2}\left(\frac{\partial T}{\partial y}\right)^2 + \left(\rho c\frac{\partial T}{\partial \theta} - Q\right) \cdot T$$

There are M subregions in R, and m refers to a particular subregion. Within each subregion R_m consider the parameters k_x, k_y, ρc and Q constant. However, these parameters may vary from subregion to subregion. As before, $\partial T/\partial\theta$ is considered invariant. It is relatively easy to show by taking a small variation of χ i.e. $\partial\chi$ for all admissible states of the variable T that the above approximates the governing partial-differential equation, the boundary conditions, and the interface conditions. Admissible states of T are defined as (i) T is continuous throughout R, (ii) the first derivatives of T are continuous in R_m, and (iii) T is equal to the specified boundary conditions on Γ_s.

Finite Element Representation of Region

The above variational principle is formally solved by the Ritz method. The Ritz method consists of selecting a trial sequence of functions that are substituted into the last equation. Such a sequence of functions is obtained by dividing R into an arbitrary number of finite elements that completely cover the domain R as illustrated in the two-dimension connected domain shown in Fig. 2. Notice that the curved boundaries are approximately modeled by straight-line segments. The triangular element is used here although quadri-

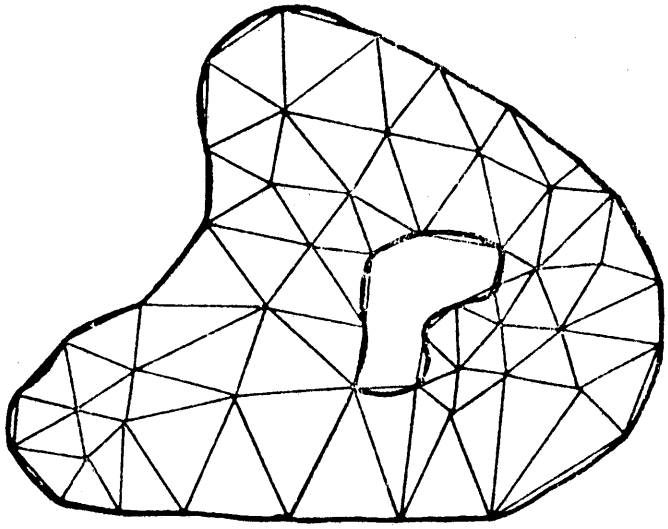


Fig. 2. Finite-element representation of connected domain

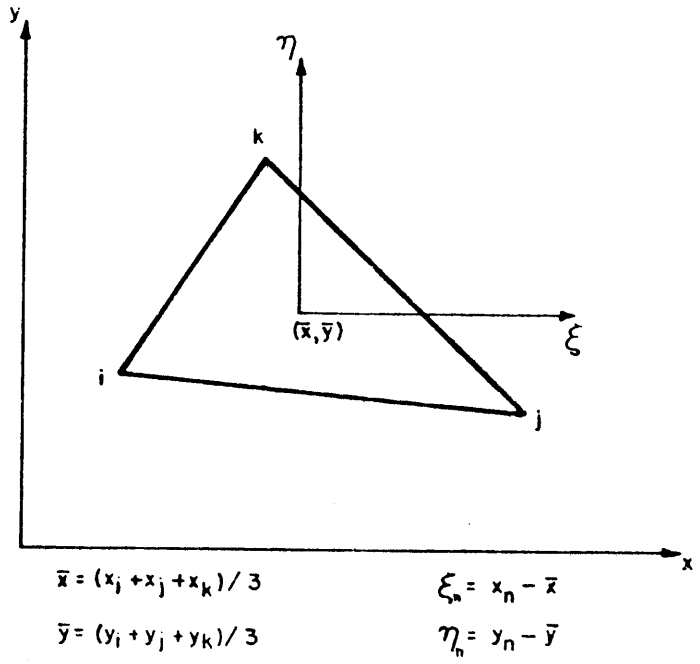


Fig. 3. General triangular element with global coordinates (x,y) and local coordinates (ξ,n)

lateral elements are more efficient from the standpoint of computer storage allocations and run times. Within each element the unknown potential states are approximated by a polynomial; for example, a linear one is used here;

$$T = B_1 + B_2 x + B_3 y$$

Admissibility requirements are met if, for a general element as shown in Fig. 3, the polynomial is forced to pass through the same value at node points which are locally designated by i, j, and k. This is simply accomplished by writing the following three equations for the potential function at each node point:

$$T_i = B_1 + B_2 x_i + B_3 y_i$$

$$T_j = B_1 + B_2 x_j + B_3 y_j$$

$$T_k = B_1 + B_2 x_k + B_3 y_k$$

and applying Cramer's rule to solve for B_1, B_2, B_3. The result for the m-th element is written in the following compact matrix notation

$$T^m = [A]\ \{T\}^m$$

The braces indicate the following column vector for the potential at each node

$$\{T\}^m = \begin{Bmatrix} T_i \\ T_j \\ T_k \end{Bmatrix}$$

The brackets indicate a row matrix for the m-th element of the form

$$[A] = [A_i, A_j, A_k]$$

The form of A_n coefficients are given in detail in Zienkiewicz (1972) and algebraic relations are given for the calculation of matrix element coefficients. These coefficients are functions of x and y, the coordinate position of the triangle nodes, and the area of the triangular element. In general, braces will indicate a column matrix and brackets will indicate a square matrix. Values of the potential state, T, are now defined in a unique and continuous manner throughout R.

To illustrate the finite-element technique, a simple polynomial is used. Greater precision can be obtained for a particular element by using a higher order polynomial.

Minimization of Variational Functional

The variational functional is minimized with respect to the potential

state T_n (n = 1, ..., N) at each node point (where there are N nodes) by evaluating each differential $\partial\chi/\partial T_n$ for each element and equating all such contributions to zero. For example, the contribution over the m-th element for the i-th node is

$$\frac{\partial \chi}{\partial T_i} = \iint_{R_m} \frac{\partial F_m}{\partial T_i} \, dxdy$$

where

$$\frac{\partial F_m}{\partial T_i} = k_x \frac{\partial T}{\partial x} \frac{\partial}{\partial T_i}\left(\frac{\partial T}{\partial x}\right) + k_y \frac{\partial T}{\partial y} \frac{\partial}{\partial T_i}\left(\frac{\partial T}{\partial y}\right) + \left(\alpha \frac{\partial T}{\partial \theta} - Q\right) \frac{\partial T}{\partial T_i}$$

The polynomial shape function is differentiated with respect to time to get

$$\partial T/\partial\theta = [\,A\,]\ \{\partial T/\partial\theta\}^m$$

This equation together with the polynomial shape function is substituted into the above to give

$$\frac{\partial F_m}{\partial T_i} = k_x \frac{\partial T}{\partial x} \frac{\partial}{\partial T_i}\left(\frac{\partial}{\partial x}[\,A\,]\{T\}^m\right) + k_y \frac{\partial T}{\partial y} \frac{\partial}{\partial T_i}\left(\frac{\partial}{\partial y}[\,A\,]\{T\}^m\right)$$

$$+ \left(\alpha [\,A\,] \left\{\frac{\partial T}{\partial \theta}\right\}^m - Q\right) \frac{\partial}{\partial T_i}[\,A\,]\ \{T\}^m$$

Since [A] is a function of x and y, constants x_n, y_n and the constant element area A_m, differentiations with respect to x and y are readily performed. Also differentiation with respect to T_i is easily performed since only $\{T\}^m$ is a function of the node state variables. Assuming parameters are constant, integration yields the desired result.

A similar operation is carried out for $\partial\chi/\partial T_j$ and $\partial\chi/\partial T_k$, and the results are combined in the following compact matrix notation

$$\{\partial\chi/\partial T\}^m = [\,s\,]\,\{T\}^m + [\,p\,]\,\{\partial T/\partial\theta\}^m - \{r\}$$

The [s] and [p] matrices, defined in numerous other references, are square, symmetrical 3 x 3 matrices that are functions only of the global coordinate position of the nodes on the m-th triangular element, the parameters for the m-th triangular element, and the area of the m-th triangular element. The {r} matrix is a function of the source Q and the area A^m.

The above equation is general in that it applies to any interior element of R provided there are no specified boundary nodes on the element. For such cases where there are specified boundary conditions, the differential of χ with respect to that node is meaningless since there is no variation. This difficulty will be resolved later.

All element system matrices are combined in accordance with the following equation

$$\frac{\partial\chi}{\partial T_n} = \sum_{m=1}^{M} \frac{\partial\chi^m}{\partial T_n}$$

which yields the following system matrix expression

$$[S^*]\ \{T\} + [P^*]\ \{\partial T/\partial\theta\} - \langle R^* \rangle$$

The brackets indicate a square symmetric matrix N X N and the braces indicate and N column matrix of the unknown potential states including the specified boundary nodes. Essentially, the above is a formal statement that all element contributions to a particular node are added together to form the equation for that node. The equation for a particular node then appears in the matrix in an ordered manner.

The above cannot yet be set to zero since the prescribed boundary conditions have not been considered. Although the boundary conditions could have been considered at the element level, they will be handled at the system level here. A scheme for handling the specified boundary conditions at the system level is given in the subsequent chapters. The starred matrices of the above are reformed by eliminating equations associated with boundary condition nodes to yield matrices [S], [P], and {R}, and the previous equation is equated to zero

$$[S]\ \{T\} + [P]\ \{\partial T/\partial\theta\} = \{R\}$$

The [S] and [P] matrices are known functions of the parameters of motion and the global coordinates of the node points. The {R} matrix is a function of the

source term and the given geometric boundary conditions. The natural boundary conditions need not be considered since they are automatically taken care of by the variational principle.

The finite-element solution is formally complete once the above has been developed. This equation is a system of linear ordinary differential equations which can be readily solved by a variety of standard methods. That is, the potential state at each unknown node point is solved for, and it is assumed that potential states vary in a linear manner between nodes. For small problems where the dimensions on the matrices are relatively small, formal integration can be used to arrive at the solution. However, for larger problems, numerical differentiation techniques are preferable.

Chapter 4

DERIVATION OF THREE-DIMENSIONAL TRANSIENT HEAT CONDUCTION MODEL WITH QUADRATIC SHAPE FUNCTION

Discussion

This chapter examines a three-dimensional transient heat problem. The only boundary conditions that will be considered are (1) given surface boundary temperatures and (2) zero heat flux on the surface boundary. The solution will be approximated by the finite-element method, using a variational statement equivalent to the governing partial-differential equation. The subject volume will be discretized into three-dimensional elements, and a quadratic interpolating shape function assumed for the field variable (temperature) in each element. The resulting information of the problem is a finite set of linear simultaneous equations, the variables of which are the values of the field variable at specific interior nodal points. Time advancement of the solution vector at nodal points is by use of the Crank-Nicolson method.

The computer program developed in this report is constructed to provide easy access to the program schemes. Modifications can be made, and further sophistications incorporated without a major overhaul of the entire program. Program schemes are separated by comment statements explaining or defining the computation algorithims and program variables, as they are derived in this report. Hence, individual processes are explained in detail by referring to this report while examining the program printout.

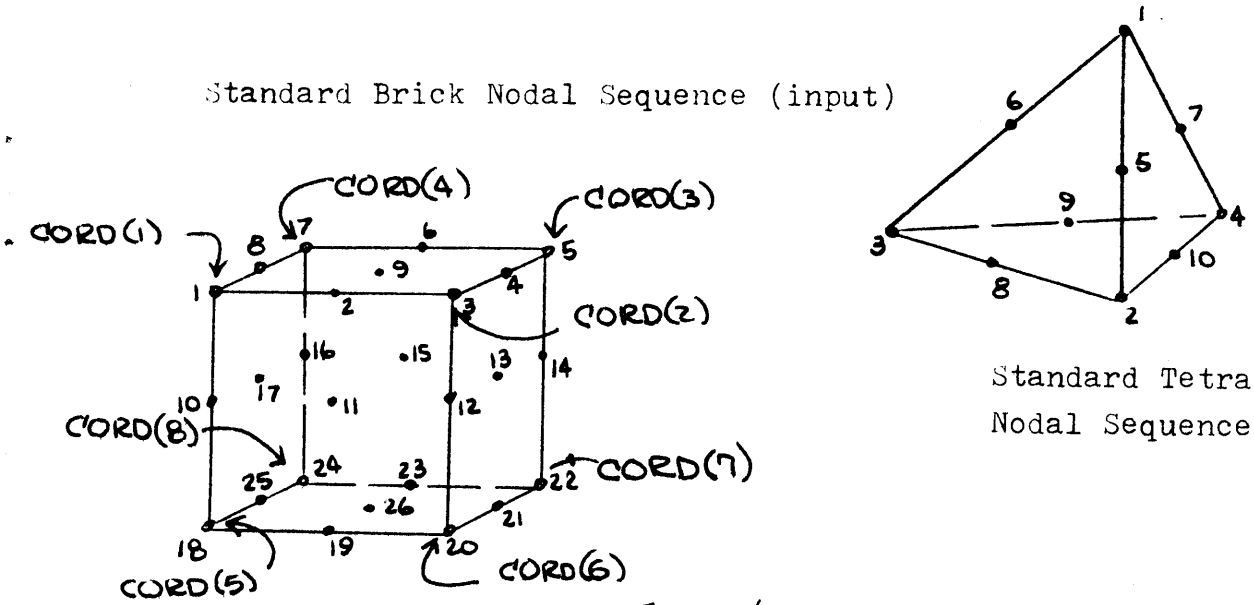
Formulation

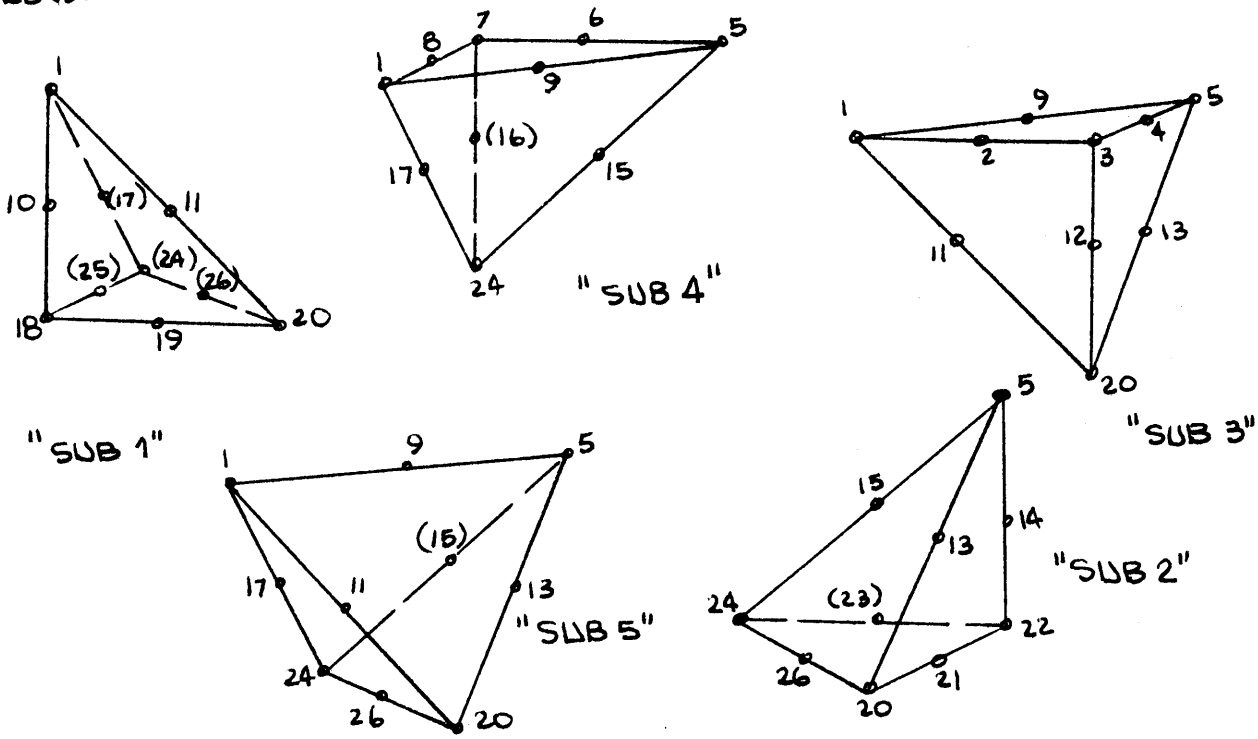
The three-dimensional volume will be discretized into tetrahedra and/or "brick" elements as shown in Fig. 4. The elements and nodal points are numbered to reduce ultimate matrix bandwidth. The element data consisting of nodal numbers and element parameters are read into the computer element by element.

The model will further subdivide the three-dimensional brick elements into five tetrahedra elements. Due to the quadratic interpolating model, there are 10 nodes per tetrahedron and 26 nodes per brick, hence the program must coordinate the bricks' nodal numbering to the five subsequent tetrahedra nodal numbering schemes. The set of five tetrahedra resulting from the subdivision of a brick element is composed of four "corner" tetrahedra of equal volumes, each being one-sixth of the brick's volume, and an interior tetrahedron having a volume of one-third of the brick's volume. Each of the five tetrahedra will assume the same intrinsic properties as ascribed to the original brick element.

The model processes the tetrahedron elements, with parameters, and constructs tetrahedron element conduction and capacitance matrices. These element matrices, each being a symmetric 10 x 10 unbanded matrix, are then incorporated into the overall global conduction and global capacitance matrices, each global matrix being formulated in "banded" form utilizing the inherent symmetry of the systems. The element matrices are directly calculated term by term using the derivations included in this report.

Finally, once all the brick's element matrices are combined into the global





Nodal Sequences	Corner Coordinate Sequence
Sub 1: 1,20,18,24,11,10,17,19,25,26	1,6,5,8
Sub 2: 5,20,24,22,13,15,14,26,23,21,	3,6,8,7
Sub 3: 5,3,1,20,4,9,13,2,11,12	3,2,1,6
Sub 4: 5,1,7,24,9,6,15,8,16,17	3,1,4,8
Sub 5: 5,1,24,20,9,15,13,17,26,11	3,1,8,6

Fig. 4. Brick Division into Five Tetrahedron

capacitance and global conduction matrix system, boundary values are inserted, the initial condition is inserted, and the time progression scheme is initiated, using the Crank-Nicolson method to move the solution vector forward with time increments of $\Delta\theta$.

Method

The volume will be assumed to be in the first quadrant of the three-dimensional coordinate system. Each node is numbered in a scheme to provide a minimum bandwidth in the banded global matrices. Element data is prepared by two methods.

Tetrahedron element - the node numbering is read into the computer in the sequence shown in Fig. 5. Any tetrahedron corner may be used as the first node entry; however, the remaining nodes must follow the illustrated sequence to ensure compatibility of the local coordinate system to the midside node numbering. Each corner's global coordinates must be read in as well as the element's parameters.

Brick element - the nodal sequence of input must follow the pattern shown in Fig. 6. It is important that the first nodal entry be as shown in the figure. Only the coordinates of the first entry are read in; the other corner coordinates are calculated in the program using the brick dimensional data. From the brick's nodal sequence, five sequences are formulated in the program for the five ultimate tetrahedron elements; hence the nodal numbering input must follow the shown pattern.

The tetradedra utilize a local volume coordinate system as discussed in Desai

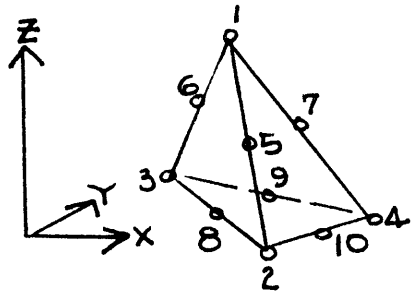


Fig. 5. Tetrahedron Numbering

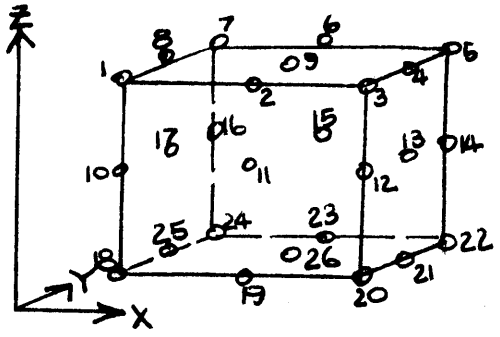


Fig. 6. Brick Numbering

and Abel. Utilizing Fig. 5, the Cartesian coordinates are related to the local coordinate system by the relation

$$\begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{Bmatrix} \text{, where } L_i = \frac{v_i}{v} \quad ;(i = 1, 2, 3, 4)$$

where each v is defined to be the volume of that tetrahedron with vertices at point (x,y,z) and the three nodes other than node "i" , $i = 1, 2, 3, 4$; and v equals the volume of the overall tetrahedron.

Define a_i to be the cofactor of x_i in the above determinant. Also define b_i and c_i to be the cofactors of y_i and z_i respectively. Then the following differentiation formulae result:

$$\frac{\partial}{\partial x} = \sum_{i=1}^{4} \frac{a_i}{6v} \frac{\partial}{\partial L_i}$$

$$\frac{\partial}{\partial y} = \sum_{i=1}^{4} \frac{b_i}{6v} \frac{\partial}{\partial L_i}$$

$$\frac{\partial}{\partial z} = \sum_{i=1}^{4} \frac{c_i}{6v} \frac{\partial}{\partial L_i}$$

where integration over the entire volume v simplifies to

$$\int_v L_1^p L_2^q L_3^r L_4^s \, dv = \frac{p!\, q!\, r!\, s!\, 6v}{(p + q + r + s + 3)!}$$

The cofactors mentioned above are derived as follows:

$$a_1 = - \begin{vmatrix} 1 & 1 & 1 \\ y_2 & y_3 & y_4 \\ z_2 & z_3 & z_4 \end{vmatrix} \quad ; \quad b_2 = - \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_3 & x_4 \\ z_1 & z_3 & z_4 \end{vmatrix} \quad ; \quad c_3 = - \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_4 \\ y_1 & y_2 & y_4 \end{vmatrix}$$

;etc.

hence,

$$a_1 = - (y_3z_4 - z_3y_4 - y_2z_4 + z_2y_4 + y_2z_3 - z_2y_3)$$

$$a_2 = + (y_3z_4 - z_3y_4 - y_1z_4 + z_1y_4 + y_1z_3 - z_1y_3)$$

$$\vdots$$

$$c_4 = + (x_2y_3 - y_2x_3 - x_1y_3 + y_1x_3 + x_1y_2 - y_1x_2)$$

Note that once the cofactors have been determined, the volume of the tetrahedron can be calculated by the expression

$$v = x_1a_1 + x_2a_2 + x_3a_3 + x_4a_4$$

However, when using a "brick" element, the volumes of the five resulting tetrahedra are, essentially, four corner tetrahedra at one-sixth the brick's volume, and one interior tetrahedron at one-third the brick's volume. Thus, using a "brick" element system easily solves the volume calculations, as compared to solving for volumes by evaluation determinants for each tetrahedron element.

Boundary values are incorporated as discussed in Myers (1971).

The global conduction and global capacitance matrix system are constructed as the tetrahedron-element contributions are determined. A "brick" element matrix system formulation is not needed.

Derivation of element "XK" and "XC" matrices

The governing partial-differential equation in a three-dimensional transient conduction heat problem on a volume v , with boundary surface v_s is

$$k_x \frac{\partial^2 t}{\partial x^2} + k_y \frac{\partial^2 t}{\partial y^2} + k_z \frac{\partial^2 t}{\partial z^2} = \rho c \frac{\partial t}{\partial \theta}$$

with boundary conditions $t(\theta) = t_o$ on the surface S , a subset of v_s and $\frac{\partial t}{\partial n}\Big|_{v_s} = 0$. The initial condition is $t(\theta = 0) = t_i$.

In this relation, the units of measurement are:

k_z, k_y, k_x = thermal conductivities in z, y, x-direction

Btu/hr - ft - ^oF

t = temperature, ^{o}F

θ = time, hours

ρ = density, lbm/ft^3

c = specific heat, $Btu/lbm - {}^{o}F$

A variational statement for this three-dimensional, transient conduction problem is the minimization of a volumetric integral over the volume v as shown below

$$I = 1/2 \iiint_v \left[k_x \left(\frac{\partial t}{\partial x}\right)^2 + k_y \left(\frac{\partial t}{\partial y}\right)^2 + k_z \left(\frac{\partial t}{\partial z}\right)^2 + \rho c \frac{\partial t^2}{\partial \theta} \right] dv \quad (6)$$

Equation 6 must be minimized for every instant in time while satisfying the boundary and initial conditions stated in Equation 1. The volumetric integral expressed in Equation 6 can be equated into the sum of four integrals

$$I = 1/2 \int_v k_x \left(\frac{\partial t}{\partial x}\right)^2 dv + 1/2 \int_v k_y \left(\frac{\partial t}{\partial y}\right)^2 dv + 1/2 \int_v k_z \left(\frac{\partial t}{\partial z}\right)^2 dv + 1/2 \int_v \rho c \frac{\partial t^2}{\partial \theta} dv \quad (7)$$

or in a different notation

$$I = I_x + I_y + I_z + I_{\rho c} \quad (8)$$

where

$$I_x = 1/2 \int_v k_x \left(\frac{\partial t}{\partial x}\right)^2 \, dv$$

and

$$I_{\rho c} = 1/2 \int_v \rho c \, \frac{\partial t^2}{\partial \theta} \, dv$$

The first step is to divide the volume v into "m" tetrahedra. Then the integral in Equation 7 is equal to the sums of the integrals over each element

$$I = \sum_{e=1}^{m} I^{(e)} = \sum_{e=1}^{m} \left(I_x^{(e)} + I_y^{(e)} + I_z^{(e)} + I_{\rho c}^{(e)} \right)$$

$$= \sum_{e=1}^{m} \left(1/2 \int_{v^e} k_x \left(\frac{\partial t}{\partial x}\right)^2 \, dv + 1/2 \int_{v^e} k_y \left(\frac{\partial t}{\partial y}\right)^2 \, dv \right.$$

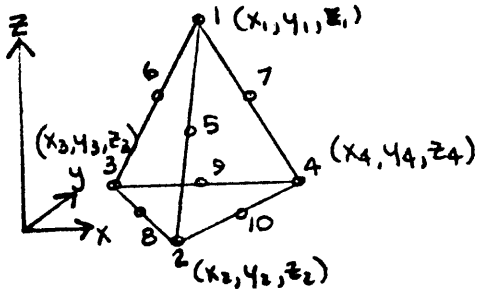
$$\left. + 1/2 \int_{v^e} k_z \left(\frac{\partial t}{\partial z}\right)^2 \, dv + 1/2 \int_{v^e} \rho c \, \frac{\partial t^2}{\partial \theta} \, dv \right) \quad (9)$$

To evaluate the integrals stated above, the field variable "t" is approximated by a quadratic polynomial. This approximation will assume "exact" values of the field variable at specified points within the element, these points are

the element's nodal points as discussed in Desai and Abel (1971) and Meyers (1971).

A typical tetrahedron element v^e with corner coordinates P_1, P_2, P_3 and P_4 is related to the local volume coordinates defined as

$$L_i = \frac{v_i}{v^{(e)}} \quad , \quad i = 1, 2, 3, 4,$$



In the quadratic approximation of the field variable

$$t = [N_1, N_2, \ldots, N_{10}]\{t_i\}$$

where the t_i are the values of temperature at the respective node numbers, and

$$[N] = \Big[\left(2L_1^2 - L_1\right)\left(2L_2^2 - L_2\right)\left(2L_3^2 - L_3\right)\left(2L_4^2 - L_4\right)\left(4L_1L_2\right)\left(4L_1L_3\right) \cdot \left(4L_1L_4\right)\left(4L_2L_3\right)\left(4L_3L_4\right)\left(4L_2L_4\right)\Big] \qquad (10)$$

Therefore, for example,

$$\frac{\partial t}{\partial x} = \frac{1}{6v^{(e)}} \sum_{i=1}^{4} a_i \frac{\partial t}{\partial L_i}$$

$$= \frac{a_1}{6v^{(e)}} \left[(4L_1 - 1)\, t_1 + 4L_2 t_5 + 4L_3 t_6 + 4L_4 t_7\right]$$

$$+ \frac{a_2}{6v^{(e)}} \left[(4L_2 - 1)\, t_2 + 4L_1 t_5 + 4L_3 t_8 + 4L_4 t_{10}\right]$$

$$+ \frac{a_3}{6v^{(e)}} \left[(4L_3 - 1)\, t_3 + 4L_1 t_6 + 4L_2 t_8 + 4L_4 t_9\right]$$

$$+ \frac{a_4}{6v^{(e)}} \left[(4L_4 - 1)\, t_4 + 4L_1 t_7 + 4L_3 t_9 + 4L_2 t_{10}\right]$$

where $\frac{\partial t}{\partial y}$, $\frac{\partial t}{\partial z}$ are related to $\frac{\partial t}{\partial x}$ except the a_i are replaced by b_i , c_i respectively; (and k_x is replaced by k_y , k_z respectively).

Substituting Equation 10 into Equation 9 transforms the transient heat conduction relation within an element "e" into a function of nodal point temperature values. This process is repeated for each element "e" of the discretized volume v. The resulting relations are combined and minimized with respect to each nodal variable, producing a system of linear equations. The boundary conditions and initial conditions are inserted. Then values for nodal

temperatures are computed at specified time step intervals by the Crank-Nicolson time advancement routine. Mathematical details are contained in Appendix A.

Chapter 5

DERIVATION OF TWO-DIMENSIONAL TRANSIENT HEAT CONDUCTION MODEL WITH QUADRATIC SHAPE FUNCTION

Discussion

This chapter examines the two-dimension transient heat conduction problem. As mentioned in Chapter 4, only zero heat flux on the surface boundary and specified nodal boundary temperatures will be considered as boundary conditions. Again, the variational principle as applied to the three-dimensional case will be utilized in two-dimensional elements, and a quadratic shape function assumed for the field variable (temperature) in each element. The resulting formulation of the problem is a finite set of linear simultaneous equations, the variables of which are the values of the field variable at specified nodal points. Time advancement of the solution vector at said nodal points is by use of the Crank-Nicolson method.

Formulation

The two-dimensional volume will be discretized into triangles and/or rectangular elements. The elements and the nodal points are numbered to reduce ultimate matrix bandwidth. The element data is read into the computer element by element as to nodal numbering and thermal parameters.

The program subdivides the two-dimensional rectangular elements into two triangles of equal volume. The quadratic shape function employs nine nodes per rectangle, and six nodes per triangle, hence the program must coordinate the rectangle's nodal

numbering to the two subsequent triangle numbering schemes. The two resulting triangles will assume the same intrinsic properties as ascribed to the two-dimensional rectangular element.

Method

The area will be assumed to be in the first quadrant of the Cartesian coordinate system. Each node is numbered in a scheme to provide a minimum bandwidth in the ultimate global matrix system. Element data is prepared by two methods:

Triangle element - the node numbering is read into the computer in the sequence shown in Fig. 7. Any triangle corner may be used as the first node entry, however, the remaining nodes must follow the illustrated sequence to ensure compatibility of the local coordinate system to the midside node numbering. Each corner's global coordinates must be read in; as well as the element's parameters.

Rectangle element - the nodal sequence of input must follow the pattern shown in Fig. 8. It is important that the first nodal entry be as shown in Fig. 8. Only the coordinates of said first entry are read in.

The triangle will utilize the local area coordinate system as discussed in Desai and Abel (1972). Utilizing Fig. 7, the Cartesian coordinates are related to the local coordinate system by the relation:

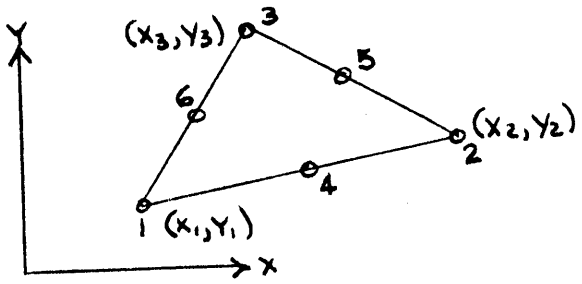


Fig. 7 Triangle Numbering

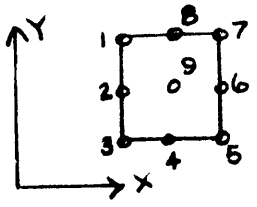


Fig. 8. Rectangle Numbering

$$\begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} , \quad L_i = \frac{A_i}{A} ; \; i = 1, 2, 3$$

where each A_i is defined to be the area of the triangles shown in Fig. 9,

and $A = A_1 + A_2 + A_3$.

Define $a_1 = x_3 - x_2$

$a_2 = x_1 - x_3$

$a_3 = x_2 - x_1$

$b_1 = y_2 - y_3$

$b_2 = y_3 - y_1$

$b_3 = y_1 - y_2$

Then the differentiation formulae for the two-dimensional case are

$$\frac{\partial}{\partial x} = \sum_{i=1}^{3} \frac{\partial L_i}{\partial x} \frac{\partial}{\partial L_i} = \sum_{i=1}^{3} \frac{b_i}{2A} \frac{\partial}{\partial L_i} ;$$

$$\frac{\partial}{\partial y} = \sum_{i=1}^{3} \frac{\partial L_i}{\partial y} \frac{\partial}{\partial L_i} = \sum_{i=1}^{3} \frac{a_i}{2A} \frac{\partial}{\partial L_i}$$

where integration over the area A results as

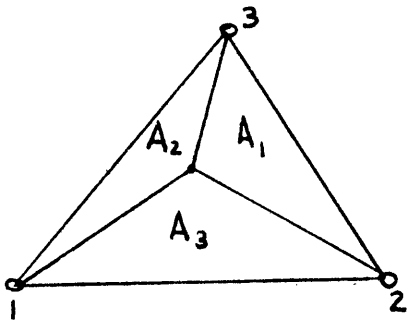


Fig. 9. Division of Triangle Element

$$\int_A L_1^p L_2^q L_3^r \, dA = \frac{p!\, q!\, r!\, 2A}{(p + q + r + 2)!}$$

Note that the area of the triangle simplifies to be

$$2A = a_3b_2 - a_2b_3 = a_1b_3 - a_3b_1 = a_2b_1 - a_1b_2$$

When utilizing rectangular elements, the triangles resulting from dividing the element have equal areas of one-half the rectangle's area.

Derivation of element "XK" and "XC" matrices

The governing partial-differential equation in the two-dimensional transient conduction heat problem on an area A, with boundary surface A_s is

$$k_x \frac{\partial^2 t}{\partial x^2} + k_y \frac{\partial^2 t}{\partial y^2} = \rho c \frac{\partial t}{\partial \theta}$$

with boundary conditions $t(\theta) = t_o$ on the surface s, a subset of A_s; and

$$\left.\frac{\partial t}{\partial n}\right|_{A_s} = 0$$

The initial condition is $t(\theta = 0) = t_i$. The derivation of the two-dimensional model parallels the derivation of the three-dimensional model (see Chapter 4), except that the z-coordinate term is omitted. Using the notation introduced in Chapter 4, let

$$I_x = 1/2 \int_A k_x \left(\frac{\partial t}{\partial x}\right)^2 dA$$

$$I_y = 1/2 \int_A k_y \left(\frac{\partial t}{\partial y}\right)^2 dA$$

and $$I_{\rho c} = 1/2 \int_A \rho c \frac{\partial t^2}{\partial \theta} dA$$

where $$I = I_x + I_y + I_{\rho c} \tag{11}$$

For a subarea A^e of the original area A, define

$$I^e = I_x^e + I_y^e + I_{\rho c}^e = 1/2 \int_{A^e} k_x \left(\frac{\partial t}{\partial x}\right)^2 dA + 1/2 \int_{A^e} k_y \left(\frac{\partial t}{\partial y}\right)^2 dA + 1/2 \int_{A^e} \rho c \frac{\partial t^2}{\partial \theta} dA \tag{12}$$

Then, if the area A is discretized into m elements, Equations 11 and 12 can be combined as

$$I = \sum_{e=1}^{m} I^e = \sum_{i=1}^{m} \left[I_x^e + I_y^e + I_{\rho c}^e \right]$$

$$= \sum_{e=1}^{m} \left[1/2 \int_{A^e} k_x \left(\frac{\partial t}{\partial x} \right)^2 dA + 1/2 \int_{A^e} k_y \left(\frac{\partial t}{\partial y} \right)^2 dA \right.$$

$$\left. + 1/2 \int_{A^e} \rho c \frac{\partial t^2}{\partial \theta} dA \right] \qquad (13)$$

a quadratic interpolating shape function is used to approximate the field variable (temperature).

In matrix notation

$$t = [N] \{t_i\} \qquad (14)$$

where t is the temperature (as a function of position within an element "e"), $\{t_i\}$ is the column vector of nodal temperatures of element "e" (in the sequence shown in Fig. 7), and $[N]$ is the row vector of shape functions defined as

$$[N] = \left[\left(2L_1^2 - L_1\right) \left(2L_2^2 - L_2\right) \left(2L_3^2 - L_3\right) \left(4L_1L_2\right) \left(4L_2L_3\right) \left(4L_1L_3\right) \right]$$

Substituting Equation 14 into Equation 13 transforms the transient heat conduction relation within an element "e" into a function of nodal point temperature values. This process is repeated for each element "e" of the discretized area A. The resulting relations are combined and minimized with respect to each nodal variable, producing a system of linear equations. The boundary conditions (specified temperatures) and initial condition are inserted. Then values for nodal temperatures are computed at specified time step intervals by the Crank-Nicolson time advancement routine. Appendix B contains further mathematical details.

Chapter 6

DERIVATION OF PHASE CHANGE APPROXIMATION

Discussion

Phase change is the major heat transport phenomena. A modified form of the procedure used by Bafus and Guymon (1975) is adopted for this model.

Consider a nodal point "A" in a discretized two- or three-dimensional domain. Phase change is not permitted to occur at node A until a requisite amount of latent heat has been exhausted or added. The prevailing quantity of latent heat for node A is determined by considering the weight of material (subject to phase transformation) of each contributing volume associated to node A in a nodal latent heat accumulator array.

As an example, consider the cooling and freezing of node A. After each time step (or a specified number of time steps), the temperature of node A is tested to determine whether or not it is within a specified tolerance of a prescribed temperature where phase transformation is assumed to occur, the freezing point depression, T_d . If the node has not been previously frozen and the computed temperature has dropped below T_d , then the quantity of latent heat evolved during the previously computed time step is

$$G = C_u \, (T_d - \text{COMPUTED TEMPERATURE}) \, (\text{VOLUME}) \, (\text{SPECIFIC WEIGHT})$$

where C_u = a weighted specific heat (at constant temperature)

Volume = volume of material assigned to node A

Specific weight = a weighted average of specific weights

The quantity G is subtracted from the latent heat accumulator for point A and, if more latent heat must still be exhausted, the computed temperature at the node is brought back to T_d prior to proceeding with the time advancement process. This is analogous to simulating phase change as an isothermal process. The time rate of latent heat generation is governed solely by the solution of the heat conduction equation and is generally insensitive to large time steps, or groups of time steps.

If the required amount or more than the required amount of latent heat has been extracted, node A and the volume of soil-water assigned to it are considered to be frozen. The thermal properties of the volume common to node A are then updated based on the volumetric proportions of soil-water and soil-ice present in the respective elements. If an excess amount of latent heat is removed, the residual is used to calculate how far below T_d the temperature at node A should be.

The thawing process is handled in a similar manner. Provisions are made to monitor which points are frozen or thawed so as to avoid refreezing frozen areas. Boundary conditions remain unaffected by the checking routines that scan point temeratures. After phase transformation at a point occurs, the latent heat accumulator assigned to that point is recomputed. Recomputation permits the simulation of long-term cyclic freezing and thawing.

Change of State

Again, consider the freezing of node A in the discretized continuum. Once the temperature of node A falls below the freezing point depression, the latent heat accumulator assigned to node A is adjusted per the amount of heat added or exhausted. Using the specific heat parameter for the unfrozen material, the amount of heat evolved in a change of temperature is

$$Q = mc_u (t_f - t_i)$$

where

m = mass (or weight, depending on units)

c_u = specific heat (unfrozen)

t_f = temperature of final state

t_i = temperature of initial state

In the English system, for example, the specific heat of water (unfrozen) is: 1 Btu/lb-^{o}F; and of ice: 0.51 Btu/lb-^{o}F . The amount of heat that must be evolved per unit mass to freeze (or thaw) water is approximately 144 Btu/ lbm or 80 cal/gm . If more than the necessary amount of latent heat is exhausted, the excess heat evolved is used to calculate the new nodal temperature, using the frozen-state specific heat at constant pressure.

The following example illustrates the above procedure:

Given that 10 lbs. of material is assigned to node A. Assume that this material is 100% water and assume that the temperature of node A actually is the average temperature of the material assigned to it. (See the next section for further discussion of average temperature). Finally, assume, for example, that the

previous time-step temperature of node A was $34^{o}F.$, and the new computed temperature is $29^{o}F.$

This change of temperature represents a heat loss of

$$Q\ (Btu) = (10\ lbs.)\ (1.00\ Btu/lb\text{-}^{o}F)\ (29^{o}F - 34^{o}F) = -50\ (Btu)$$

However, node A cannot assume $29^{o}F$ until the requisite (144 Btu/lb) (10 lb) = 1,440 Btu of latent heat is exhausted. Hence, node A would be set at $32^{o}F$, and the latent heat accumulator reduced as follows:

$$\text{LATENT HEAT (NODE A)} = 1,440 + (10)\ (1)\ (29^{o}F - 32^{o}F)$$

$$= 1,440 - 30$$

$$= 1,410$$

Consider that the latent heat accumulator for node A is almost zero, perhaps 10 Btu, and the new computed temperature for node A is, for example, $25^{o}F.$ Then we proceed as follows:

$$Q = (10\ lbs)\ (1.00\ Btu/lb - {}^{o}F)\ (32^{o}F - 25^{o}F)$$

$$= 70\ Btu,$$ which is greater than the remaining latent heat for node A This implies the material assigned to node A is now frozen, and yet 60 Btu of heat remains to be accounted for. A new temperature has to be calculated, where a change of temperature is a function of the specific heat of ice, at 0.51 Btu/lb-^{o}F, and the amount of heat involved. Hence, the new temperature of node A is computed:

$$60\ Btu = (10\ lbs.)\ (0.51\ Btu/lb\text{-}^{o}F)\ (32^{o}F - t_{f})$$

Thus, $11.76 = 32 - t_{f}$ or $t_{f} = (32 - 11.76)\ ^{o}F = 20.24^{o}F$

The latent heat accumulator for node A is now zero, and the time progression

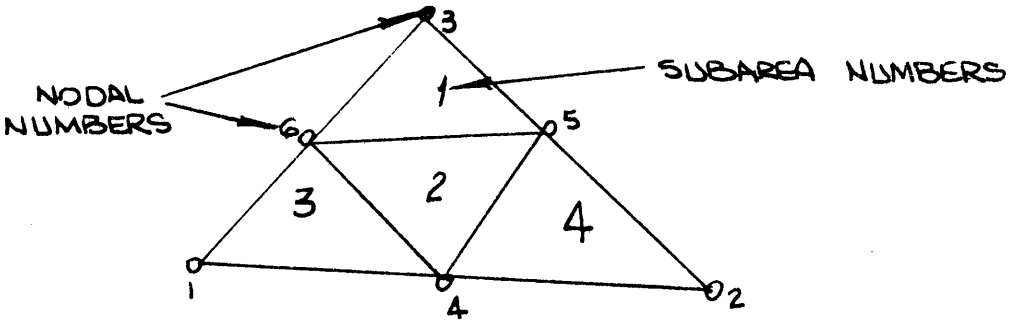
continues, but with node A now set initially at 20.24°F . Note how the phase change dominates the process of lowering the temperature of node A from 34°F to 20.24°F.

Average Temperature - Two Dimensions

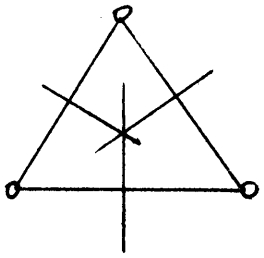
Each triangular element contains six nodes. Each node is a sample temperature within the region of material associated to it. It will be shown how the material is associated to each node, and how the nodal temperature is used to determine phase change.

Each triangular element can be considered as the union of four triangles, each containing three nodal points. The boundaries of the geometry associated to the nodes are determined by the perpendicular bisectors of the sides of the interior triangles (see Fig. 10). The model makes the following assumption: When a nodal temperature is within a specified tolerance of the phase change temperature, the nodal temperature is assumed to represent the average temperature of all the associated material to the node, and this temperature is then used for phase change determination as discussed in the previous section. But is this simplification valid? The following discussion approaches the problem by a rigorous interpretation.

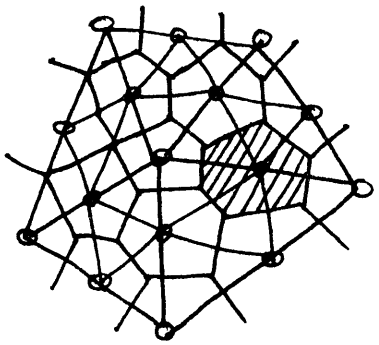
Consider the freezing of node A, as illustrated by the hatched geometry in Fig. 10, and a computer model utilizing exact temperature. The computer program keeps track of each element containing node A. Once node A is tested for phase change, each element determined as containing node A is scanned as to the



A quadratic element is considered the union of four interior triangles-each with three nodal points.



Each interior triangle is subdivided by the extensions of the perpendicular bisectors of each side of the triangle



A continuum subdivided into the geometries associated to each node for phase change study.

Fig. 10. Phase Change Scheme

location of node A within the element (i. e. as to whether node A is a midside node or a corner node). The area of each element associated to node A by the geometric subdivision through perpendicular bisection (Fig. 10) of the triangle sides is calculated, then multiplied by the average temperature associated to this sub-area, and then this product or weighted contribution is added to the similar weighted contribution for each element associated to node A.

This sum of weighted contributions is divided by the total area associated to node A, resulting in the average temperature of the geometry assigned to node A.

Consider now the average temperature of an entire quadratic triangular element in our two-dimensional problem. The average temperature for the entire area is simply

$$\bar{t}_e = 1/A_e \int_{A_e} [N]\ \phi_i dA$$

where A is the area of the element, [N] is the interpolating shape function matrix, ϕ_i is the vector of nodal temperatures.

From a previous section, we have

$$A\bar{t}_e = \int_{A_e} (2L_1^2 - L_1)\ \phi_1 dA + \int_{A_e} (2L_2^2 - L_2)\phi_2 dA + \int_{A_e} (2L_3^2 - L_3)\ \phi_3 dA$$

$$+ \int_{A_e} 4L_1 L_2 \phi_4\, dA + \int_{A_e} 4L_2 L_3 \phi_5\, dA + \int_{A_e} 4L_3 L_1 \phi_6\, dA$$

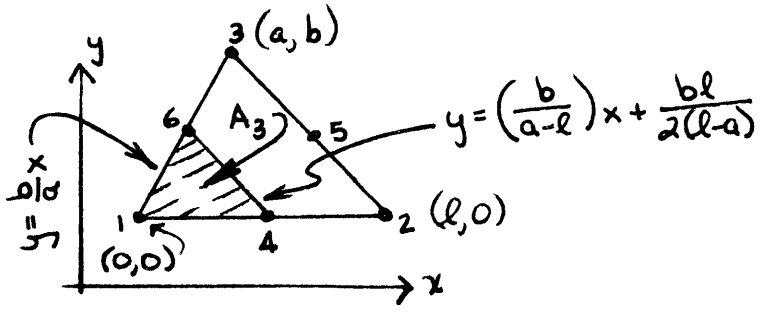
$$= A/3 \quad (\phi_4 + \phi_5 + \phi_6)$$

$$\therefore \bar{t}_e = 1/3 \ (\phi_4 + \phi_5 + \phi_6)$$

the average of the midside nodes. Consider the average temperatures of each subtriangle in the top of Fig. 10. The sum of the average temperatures, multiplied by their respective area, should ultimately sum up to be equal to $4A\bar{t}_e$,

$$\text{i.e.,} \quad 1/4A \sum_{i=1}^{4} A_i \bar{t}_{A_i} = \bar{t}_e = 1/3 \ (\phi_4 + \phi_5 + \phi_6)$$

We shall now derive these average temperatures, $\bar{t}A_i$, i = 1, 2, 3, 4 . Consider the element oriented as shown below:



the local coordinate $L_3 = \frac{(1/2)\ell y}{(1/2)\ell b} = y/b$

$$\therefore 2L_3^2 - L_3 = 2y^2/b^2 - y/b$$

then the integration for ϕ_3 results in

$$A_3\bar{t}_{A_3} = \int_0^{a/2}\int_0^{bx/a}\left(\frac{2y^2}{b^2} - \frac{y}{b}\right)dydx + \int_{a/2}^{\ell/2}\int_0^{\left[\left(\frac{b}{a-\ell}\right)x + \frac{b\ell}{2(\ell-a)}\right]}\left(\frac{2y^2}{b^2} - \frac{y}{b}\right)dydx$$

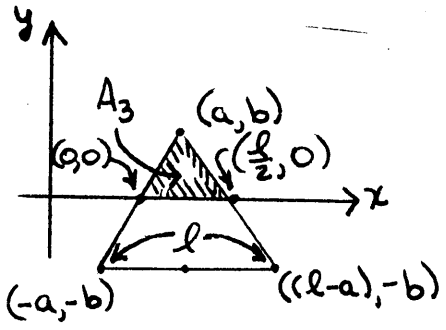
$$(\phi_3)$$

$$= \frac{-b\ell}{96}, \quad A_3 = (1/2)(\ell/2)(b/2) = b\ell/8$$

$$\therefore \{\phi_3\}\,\bar{t}_{A_3} = \left(\frac{-b\ell}{96}\right)\left(\frac{8}{b\ell}\right)\phi_3 = -\frac{1}{12}\phi_3$$

By argument that the triangle is arbitrarily oriented, the coefficient of ϕ_2 must also be - 1/12 .

Consider a new orientation of the element:



Then the integration for ϕ_1 results in:

$$A_3 \bar{t}_{A_3} = \int_0^a \int_0^{bx/a} (2\left(\frac{y}{2b} + \frac{1}{2}\right)^2 - \left(\frac{y}{2b} + \frac{1}{2}\right)) \, dydx$$
$$(\phi_1)$$

$$+ \int_a^{\ell/2} \int_0^{\frac{2bx-b\ell}{2a-\ell}} (2\left(\frac{y}{2b} + \frac{1}{2}\right)^2 - \left(\frac{y}{2b} + \frac{1}{2}\right)) \, dydx$$

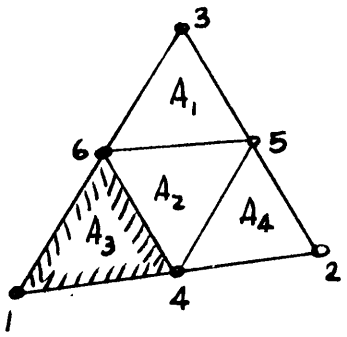
$$= 3b\ell/48$$

where $A_3 = (\ell/2)\ (b/2) = b\ell/4$

$$\therefore \{\phi_1\}\ \bar{t}\ A_3 = \left(\frac{3b\ell}{48}\right)\left(\frac{4}{b\ell}\right)\phi_1 = \frac{3}{12}\phi_1$$

So, for an arbitrary corner subtriangle of a quadratic triangular element we have intermediate values for the average temperature:

$$\bar{t}_{A_3} = 3/12\ \phi_1 - 1/12\ \phi_2 - 1/12\ \phi_3 + \beta_1\phi_4 + \beta_2\phi_5 + \beta_3\phi_6$$



By analogy that A_1 and A_4 are also corner subtriangles,

$$\bar{t}_{A_1} = -1/12\ \phi_1 - 1/12\ \phi_2 + 3/12\ \phi_3 + \alpha_1\phi_4 + \alpha_2\phi_5 + \alpha_3\phi_6$$

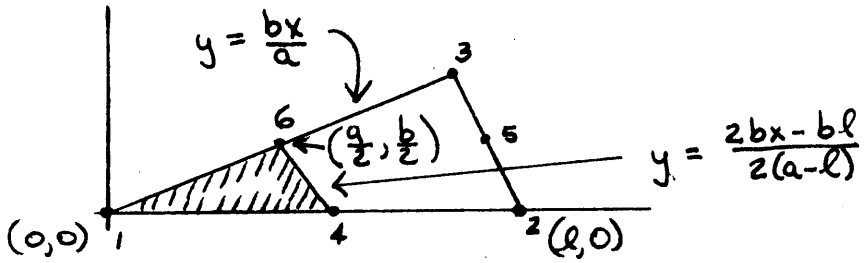
$$\bar{t}_{A_4} = -1/12\ \phi_1 + 3/12\ \phi_2 - 1/12\ \phi_3 + \gamma_1\phi_4 + \gamma_2\phi_5 + \gamma_3\phi_6$$

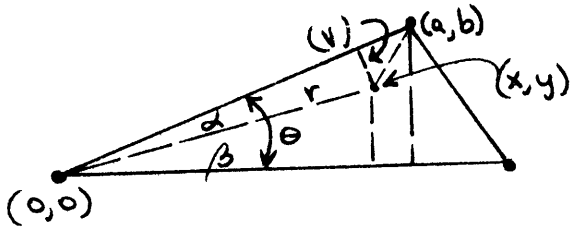
But since $\bar{t}_e = 1/3\ (\phi_4 + \phi_5 + \phi_6)$ we can deduce that

$$\bar{t}_{A_2} = -1/12\ \phi_1 - 1/12\ \phi_2 - 1/12\ \phi_3 + \eta_1\phi_4 + \eta_2\phi_5 + \eta_3\phi_6$$

By the previous work, we conclude that an arbitrarily formed triangle can be rotated for midside nodal integrations; hence, we deduce that $\eta_1 = \eta_2 = \eta_3$. Therefore, $\eta_i = 5/12$, $i = 1, 2, 3$, (also note that $\sum_{i=1}^{6}$ (coefficients)ϕ_i=1 for each subtriangle).

Consider the following orientation of the quadratic triangular element, and the subsequent local coordinate calculations:



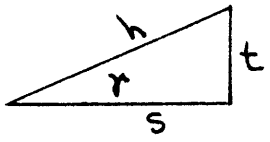


$\tan(\theta) = b/a$

$\tan(\beta) = y/x$

$\theta = \alpha + \beta \Rightarrow \alpha = \theta - \beta$

$$\therefore \tan(\alpha) = \tan(\theta - \beta) = \left\{ \frac{\frac{b}{a} - \frac{y}{x}}{1 + \frac{by}{ax}} \right\} = \frac{bx - ay}{ax + by}$$



$\tan(\gamma) = t/s$ and $h^2 = t^2 + s^2$

$\therefore \sin(\gamma) = t/h$, where $v = r \sin(\alpha)$

$\Rightarrow v = (x^2 + y^2)^{1/2} \sin(\alpha)$

$$= \frac{(x^2 + y^2)^{1/2} (bx - ay)}{\left[(bx - ay)^2 + (ax + by)^2\right]^{1/2}}$$

$$= \frac{(x^2 + y^2)^{1/2} (bx - ay)}{\left[x^2 (b^2 + a^2) + y^2 (b^2 + a^2)\right]^{1/2}}$$

$$= \frac{(x^2 + y^2)^{1/2}(bx - ay)}{\left[(a^2 + b^2) \cdot (x^2 + y^2)\right]^{1/2}} = \frac{bx - ay}{(a^2 + b^2)^{1/2}}$$

$$\therefore \quad L_2 = \frac{1/2 \quad (a^2 + b^2)^{1/2} \left\{ \frac{(bx - ay)}{(a^2 + b^2)^{1/2}} \right\}}{1/2 \; \ell b} = \frac{bx - ay}{b\ell} \; ,$$

$$\text{and} \quad L_3 = \frac{(1/2)\ell y}{(1/2)\ell b} = \frac{y}{b}$$

$$\therefore A_3 \bar{t}_{A_3(\phi_5)} = 4 \iint_{A_3} L_2 L_3 \, dA = 4 \iint_{A_3} \frac{bxy - ay^2}{b^2 \ell} \, dA, \text{ where } A_3 = \frac{b\ell}{8} \text{ (by diagram)}$$

$$\Rightarrow 4/A_3 \int_0^{a/2} \int_0^{bx/a} \left(\frac{xy}{b\ell} - \frac{ay^2}{b^2 \ell} \right) dy dx + 4/A_3 \int_{a/2}^{\ell/2} \int_0^{\left(\frac{2bx - b\ell}{2(a-\ell)}\right)} \left(\frac{xy}{b\ell} - \frac{ay^2}{b^2 \ell} \right) dy dx$$

$$= 1/12$$

Thus, all midside nodes opposite the subtriangle being investigated have a temperature contribution of 1/12 , and by previous arguments of orientation,

$$\alpha_1 = \beta_2 = \gamma_3 \; ; \; \alpha_2 = \alpha_3 = \beta_1 = \beta_3 = \gamma_1 = \gamma_2$$

Thus, $\alpha_1 = 1/12$ and $\alpha_2 = 5/12$.

$$\therefore \; \bar{t}_{A_1} = -1/12\,\phi_1 - 1/12\,\phi_2 + 3/12\,\phi_3 + 1/12\,\phi_4 + 5/12\,\phi_5 + 5/12\,\phi_6$$

$$\bar{t}_{A_2} = -1/12\,\phi_1 - 1/12\,\phi_2 - 1/12\,\phi_3 + 5/12\,\phi_4 + 5/12\,\phi_5 + 5/12\,\phi_6$$

$$\bar{t}_{A_3} = 3/12\,\phi_1 - 1/12\,\phi_2 - 1/12\,\phi_3 + 5/12\,\phi_4 + 1/12\,\phi_5 + 5/12\,\phi_6$$

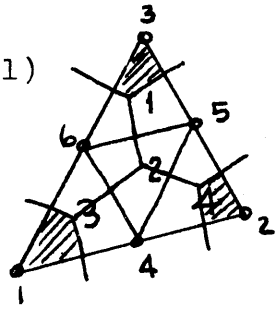
$$\bar{t}_{A_4} = -1/12\,\phi_1 + 3/12\,\phi_2 - 1/12\,\phi_3 + 5/12\,\phi_4 + 5/12\,\phi_5 + 1/12\,\phi_6$$

where $$1/4 \sum_{i=1}^{4} \bar{t}_{A_i} = \bar{t}_e = 1/3\,(\phi_4 + \phi_5 + \phi_6)$$

By geometry, each $A_i = A/4$, for A = area of quadratic element. We will assume that $(1/3)A_i$ is contributed to each of the three nodes attached to A_i , thus a contribution of $A_e/12$ is weighted to each of the nodes per each associated subtriangle.

We thus standardize the weighted contributions of the overall quadratic element per each of the six nodes, letting W.C. (i) = weighted contribution for node (i):

(1)



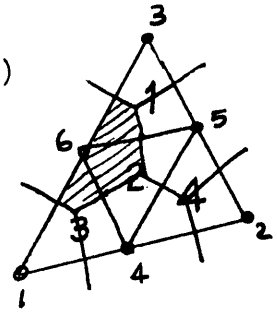
each corner node ϕ_1, ϕ_2, ϕ_3

W.C. (i) = A/12 $\bar{t}_{A_i}$, i = 1, 2, 3

where A_i is that subtriangle

associated to node "i"

(2)



each midside node ϕ_4, ϕ_5, ϕ_6

W.C. (4) = A/12 ($\bar{t}_{A_3} + \bar{t}_{A_4} + \bar{t}_{A_2}$)

W.C. (5) = A/12 ($\bar{t}_{A_2} + \bar{t}_{A_4} + \bar{t}_{A_1}$)

W.C. (6) = A/12 ($\bar{t}_{A_1} + \bar{t}_{A_2} + \bar{t}_{A_3}$)

or equivalently,

W.C. (4) = A/12 (1/12 ϕ_1 + 1/12 ϕ_2 - 3/12 ϕ_3 + 15/12 ϕ_4 + 11/12 ϕ_5 + 11/12 ϕ_6)

W.C. (5) = A/12 (-3/12 ϕ_1 + 1/12 ϕ_2 + 1/12 ϕ_3 + 11/12 ϕ_4 + 15/12 ϕ_5 + 11/12 ϕ_6)

W.C. (6) = A/12 (1/12 ϕ_1 + -3/12 ϕ_2 + 1/12 ϕ_3 + 11/12 ϕ_4 + 11/12 ϕ_5 + 15/12 ϕ_6)

where $\sum_{i=1}^{6}$ W.C. (i) $= A_e/12 \ (48/12 \ \phi_4 + 48/12 \ \phi_5 + 48/12 \ \phi_6)$

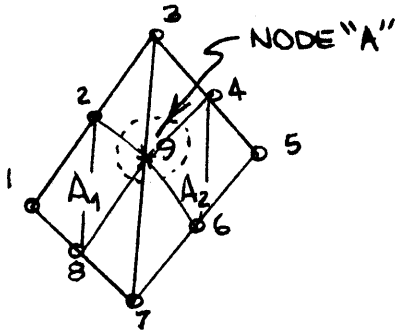
$= A_e/3 \ (\phi_4 + \phi_5 + \phi_6) = A_e \bar{t}_e$

which agrees with the above derivations of $\bar{t}_e$.

We shall use the above derivations to support the simplification of letting the calculated value of node "i" be the average temperature of the associated area to node "i" .

"Aspect Ratio" reasoning dictates that element diameter ratios should be as closed to 1 - 1 as possible. Thus, one trianglular element would be of the same approximate shape and size as its neighboring elements; otherwise errors would be introduced in the exaggerated element's direction (see Desai and Abel (1972)).

Consider an arbitrary node "i" and its associated area (Fig. 10). The average temperature is the sum of all weighted contributions divided by the total contributed area. By the above, the temperature at node "A" approximates the average temperature for the associated area, hence eliminating the above calculations. The reasoning follows for a simple midside nodal calculation:



let $A_1 \approx A_2$,

$$\frac{(A_1 + A_2)}{4} \bar{t}_{"A"} = A/12 \left(-3/12 [\phi_1 + \phi_5] + 2/12 [\phi_7 + \phi_3]\right.$$

$$\left. + 11/12 [\phi_2 + \phi_6 + \phi_4 + \phi_8] + 30/12 [\phi_9]\right)$$

If the triangles are similar, we can see each bracketed sum approximates ϕ_9 in the average, hence

$$\left(\frac{A_1}{4} + \frac{A_2}{4}\right) \bar{t}_{"A"} \approx \frac{A}{2} \bar{t}_A = \frac{A}{12} (6\phi_9) = \frac{A}{2} \phi_9$$

which implies that $\bar{t}_{"A"} \approx \phi_9$.

Thus, the more exacting computer model (which employs temperature averaging of all contributing elements) can be simplified by letting the nodal temperature represent the average temperature of the material associated to the node. This saves a significant portion of the computer time involved in a phase change process, and eliminates the necessity of storing the nodal locations of each associated element.

Computer Simulation of Element Phase Change

The phase change process is modeled in the computer program by updating the thermal parameters of any element whose nodes have all experienced the

transformation of either freezing or thawing. Noting that the phase change portion of the program is restricted to isotropic (thermal) domains, the thermal conductivities in the x, y and z directions are all equal to the parameter designated as "k_u" or "k_f" , i.e. the thermal conductivity for unfrozen and frozen material respectively. Thus, once a node freezes, each element associated with the newly frozen node is tested to determine whether all the nodes in the element are now frozen; if so, then the parameters are are updated for that element, if not, the program continues without an element phase change modification.

To modify the global matrices for the changed thermal conductivity parameter of a newly frozen element, the difference $(k_f - k_u)$ is calculated and then used as the thermal conductivity of the newly frozen element, and the element is reprocessed in the element matrix derivation subroutine. This technique alters the global matrix system without revising the entire global matrix system element by element. For a newly thawed element, the above procedure is paralleled, but with the element's thermal conductivity parameter adjusted by the difference $(k_u - k_f)$. See Appendix C for "global matrix phase change updating" derivation.

Chapter 7

CRANK-NICOLSON METHOD

Once the volumetric brick/tetrahedron elements have been amalgamated into the global matrices "$\underset{\sim}{GK}$" and "$\underset{\sim}{GC}$", where "$\underset{\sim}{GK}$" is the global conduction matrix and "$\underset{\sim}{GC}$" is the global capacitance matrix, the problem becomes one of solving the following system of linear first order differential equations:

$$\frac{dI}{dt} = \underset{\sim}{GK}t + \underset{\sim}{GC}\dot{t} = 0 ,$$

or

$$\underset{\sim}{GC}\dot{t} = -\underset{\sim}{GK}t$$

where $\dot{t}$ and t represent the time differential of the field variable vector, and the field variable vector , respectively.

The Crank-Nicolson method moves the solution vector ahead in time intervals of $\Delta\theta$ by the relation

$$t^{(v+1)} = t^{(v)} + \frac{\Delta\theta}{2}\left(\dot{t}^{(v)} + \dot{t}^{(v+1)}\right) ,$$

or equivalently

$$\underset{\sim}{GC}t^{(v+1)} = \underset{\sim}{GC}t^{(v)} + \frac{\Delta\theta}{2}\left(\underset{\sim}{GC}\dot{t}^{(v)} + \underset{\sim}{GC}\dot{t}^{(v+1)}\right) ,$$

where

$$\underset{\sim}{GC}\dot{t}^{(i)} = - \underset{\sim}{GK}t^{(i)}$$

Hence,

$$\underset{\sim}{GC}t^{(v+1)} = \underset{\sim}{GC}t^{(v)} + \frac{\Delta\theta}{2}\left(-\underset{\sim}{GK}t^{(v)} - \underset{\sim}{GK}t^{(v+1)}\right)$$

Combining terms,

$$\left(\underset{\sim}{GC} + \frac{\Delta\theta}{2}\underset{\sim}{GK}\right)t^{(v+1)} = \left(\underset{\sim}{GC} - \frac{\Delta\theta}{2}\underset{\sim}{GK}t^{(v)}\right)$$

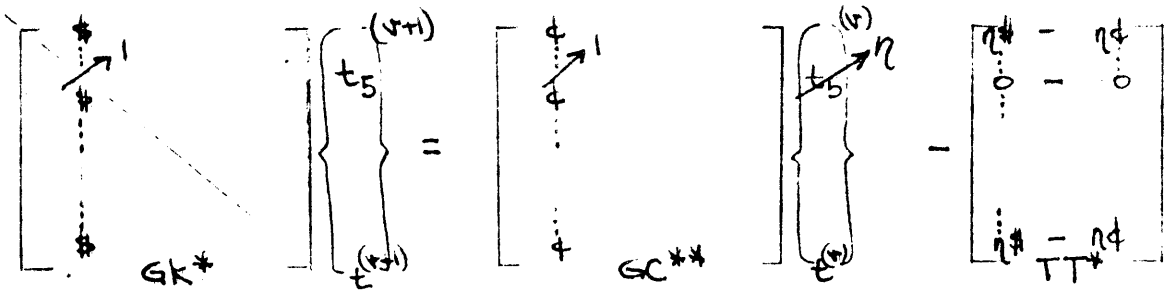
where for

$$\Delta\theta > 0\ , \qquad \left(\frac{2}{\Delta\theta}\underset{\sim}{GC} + \underset{\sim}{GK}\right)t^{(v+1)} = \left(\frac{2}{\Delta\theta}\underset{\sim}{GC} - \underset{\sim}{GK}\right)t^{(v)} \qquad (1)$$

The global matrices "$\underset{\sim}{GK}$" and "$\underset{\sim}{GC}$" are modified to represent Equation 1 without additional storage requirements. By letting the matrix GC* equal $\frac{2}{\Delta\theta}\underset{\sim}{GC}$, we can form System 1 by setting $\underset{\sim}{GK}^* = \underset{\sim}{GK} + \underset{\sim}{GC}^*$; $\underset{\sim}{GC}^{**} = 2\underset{\sim}{GC}^* - \underset{\sim}{GK}^*$, where in each new relation, the new forms of the GC and GK matrices are used as they are produced. Hence, the System 1, modified as above, reads as

$$\underset{\sim}{GK}^*t^{(v+1)} = \underset{\sim}{GC}^{**}t^{(v)} \qquad (2)$$

It is this form of Equation 1 that is directly formed in the computer. Before solution of Equation 2, the boundary conditions (specified temperatures) are inserted by the process described in Meyers (1971): let $t_5 = \eta$, a specified boundary temperature, then for Equation 2:



where the "$" represent column terms in the $\underset{\sim}{GK}^*$ matrix, and the "¢" represent column terms in the $\underset{\sim}{GC}^{**}$ matrix.

(1) set $\underset{\sim}{GK}^*(5,5) = \underset{\sim}{GC}^{**}(5,5) = 1.0$

(2) set $\underset{\sim}{TT}^*(I) = \eta(\$ - ¢)$, $I \neq 5$; $-\underset{\sim}{TT}^*(I) = \underset{\sim}{TT}(I)$

(3) set "5" row and "5" column of $\underset{\sim}{GK}^*$ & $\underset{\sim}{GC}^{**} = 0.0$, except for statement (1).

Now the system $\underset{\sim}{GK}t^{(v+1)} = \underset{\sim}{GC}t^{(v)} + \underset{\sim}{TT}$, where $\underset{\sim}{GK}$, $\underset{\sim}{GC}$, and $\underset{\sim}{TT}$ represent the modified matrices of Equation 2 (but with inserted boundary conditions) is ready for a time-advancement solution scheme with specified time steps.

Chapter 8

MODEL EVALUATION

Variables of Convergence

The computer program utilizes a quadratic shape function of the finite element approximation for temperature (field variable) in a three-dimensional or two-dimensional continuum. As discussed in Myers (1971), the solution oscillates about the analytic solution before convergence. A balance of element size, shape time step size, and the magnitude of the difference between the initial temperature and boundary temperatures is required for an optimum solution to the problem.

Thirty runs were made varying individually the above mentioned parameters on a control problem. A basic transient heat-conduction problem was employed and compared to the analytic solution by varying the (1) size and shape of the elements; (2) the time step increments and (3) the temperature "shock" to the system.

Convergence Analysis

A six foot width "plane wall" problem was examined for the following cases:

(1) Approximate the system by six cubic elements of one foot dimensions. Use a three-dimensional model. Shock the system by a temperature difference of 200°F. Use a time step of 0.1 hours. The results were considered "converged" after approximately 5 hours into the solution. A two-dimensional approximation, with similar parameters, was conducted and revealed similar answers. Thus, the two-dimensional case was used for the remainder of the testing. The necessity for quick convergence to the exact solution is primarily due to the solution deviations caused by phase changes at incorrect times. Hence, if convergence criteria is established for large temperature "shocks" to the system, convergence is assured for the anticipated temperature differences imposed in a more natural situation, such as 5°F, more

or less. Additionally, sudden "freezes," as may occur in chilled climates with sudden high velocity winds, will not destroy the model. Two shocks for the three-dimensional system were used, a 100°F and 200°F temperature change. The two-dimensional model yielded the same results in both tests.

(2) Using the two-dimensional model, a temperature "shock" of two hundred degrees Fahrenheit, and an element size of 1 foot dimensions (square), vary the time step increments letting "theta" equal 0.05, 0.10, 0.25, 0.50, 1.00, 2.50, 5.0, 10.0 and 24.0 hours.

(3) Using the two-dimensional model, a temperature "shock" of two hundred degrees Fahrenheit, and time steps of 0.05, 0.1, 0.25, 0.50, 1.00, 2.50 and 5.0 hours vary the size of elements (square) at 1 foot, 1.5 feet, and 3.00 feet.

Analytic Solution

The problem considered is actual a one-dimensional "plane-wall" problem. The wall is assumed to be iron (isotropic with thermal conductivity of 31.4 Btu/hr ft-°F; density of 492 lbm/ft^3; heat capacity of 0.137 Btu/lbm-°F.

The governing differential equation is

$$k_x \frac{\partial^2 t}{\partial x^2} = \rho c \frac{\partial t}{\partial \theta}$$

where the boundary conditions are that $t = t_\infty$ @ $x = 0,6$. The initial condition is $t(x, \theta=0) = t_i$. The problem is normalized using the variables

$$x^* = \frac{x}{6} \qquad \theta^* = \frac{\alpha\theta}{36} \qquad t^* = \frac{t-t_\infty}{t_i-t_\infty} \qquad \alpha = \frac{k_x}{\rho c}$$

Then the differential equations simplifies to:

$$\frac{\partial^2 t^*}{\partial (x^*)^2} = \frac{\partial t^*}{\partial \theta^*} \; ; \; t^*(0,\theta^*) = 0, \; t^*(1,\theta^*) = 0, \; t^*(x^*,0) = 1$$

The solution is the power series:

$$t^*(X^*,\theta^*) = \frac{4}{\pi} \left(\sin x^* e^{-\pi^2\theta^*} + \frac{1}{3} \sin 3\pi x^* e^{-9\pi^2\theta^*} + \frac{1}{5} \sin 5\pi x^* e^{-25\pi^2\theta^*} + \ldots\right)$$

Solutions to this equation for the case where the boundary condition temperature at x = 0 is 100°F and at x = ∞ is - 100°F are shown in Table 1.

Test Results

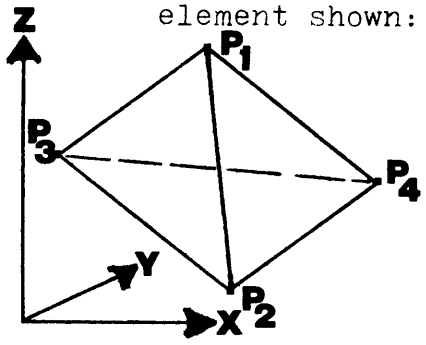
The model output was examined as to when convergence occurred. The time (in model simulation) of convergence was plotted graphically as a function of time-step size and element size. For the Transient Heat Conduction model, a time step size of 0.1 hours, and the convenient element size of 1.0 foot was found adequate in rate of convergence. Results are shown in Figures 11, 12 and 13.

3-D Test Problem

The local coordinate relation to global coordinates is stated in Desai and Abel (1972) as

$$\begin{Bmatrix} 1 \\ X \\ Y \\ Z \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ X(1) & X(2) & X(3) & X(4) \\ Y(1) & Y(2) & Y(3) & Y(4) \\ Z(1) & Z(2) & Z(3) & Z(4) \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix}$$

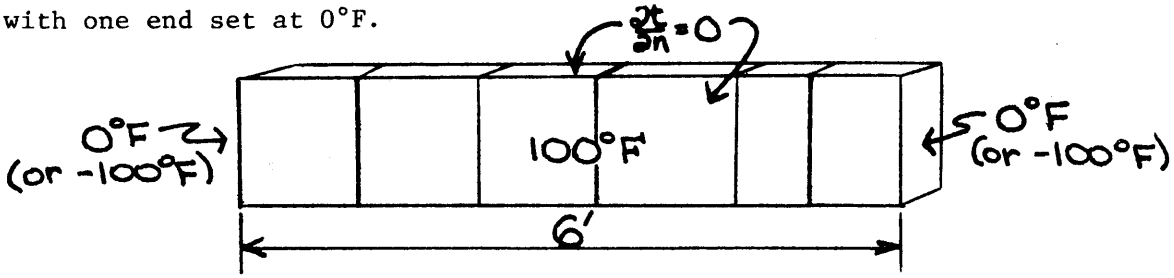
for the tetrahedron element shown:



The corner coordinates are numbered by choosing any vertex as P_1 (X,Y,Z), and numbering the remaining three corners in a clock-wise direction as viewed from P_1. The coordinates of the P_i are then substituted into the above matrix.

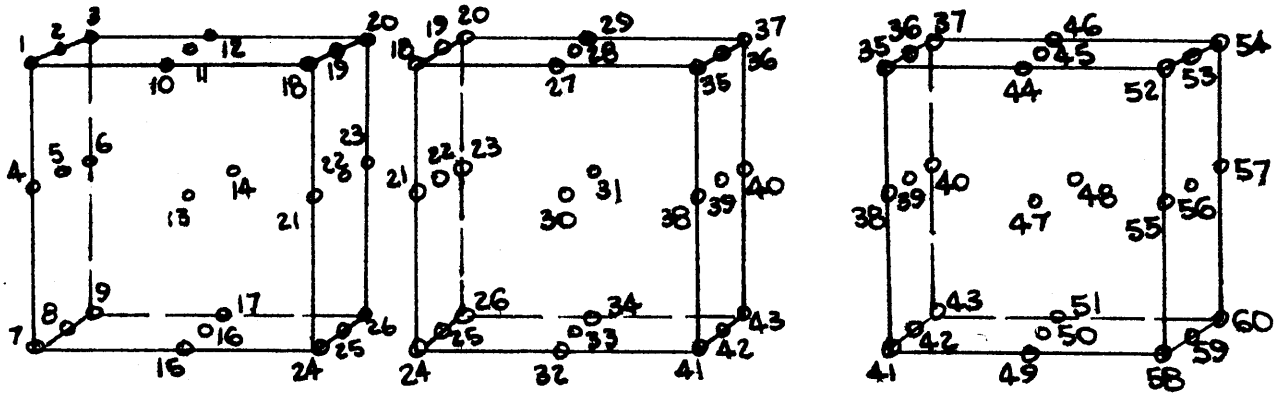
The coordinate system is established in the figure. Note that the y-coordinate is positive into the paper. Thus, the volume under consideration must be dimensional from the origin shown above. The problem tested in the

routine is an insulated iron homogeneous isotropic bar. The bar is six feet long, with a square cross section of one-ft^2 area. The bar is initially assumed to be at 100°F. Then, the ends are both set at 0°F. Due to symmetry, the problem is simplified to an insulated bar of 1 ft^2 cross sectional area, insulated at one end and along the surface boundary, initially set at 100°F, with one end set at 0°F.



The volume is discretized into 3 cubic brick elements of equal volume $V^{(e)}$.

The elements are formed below:



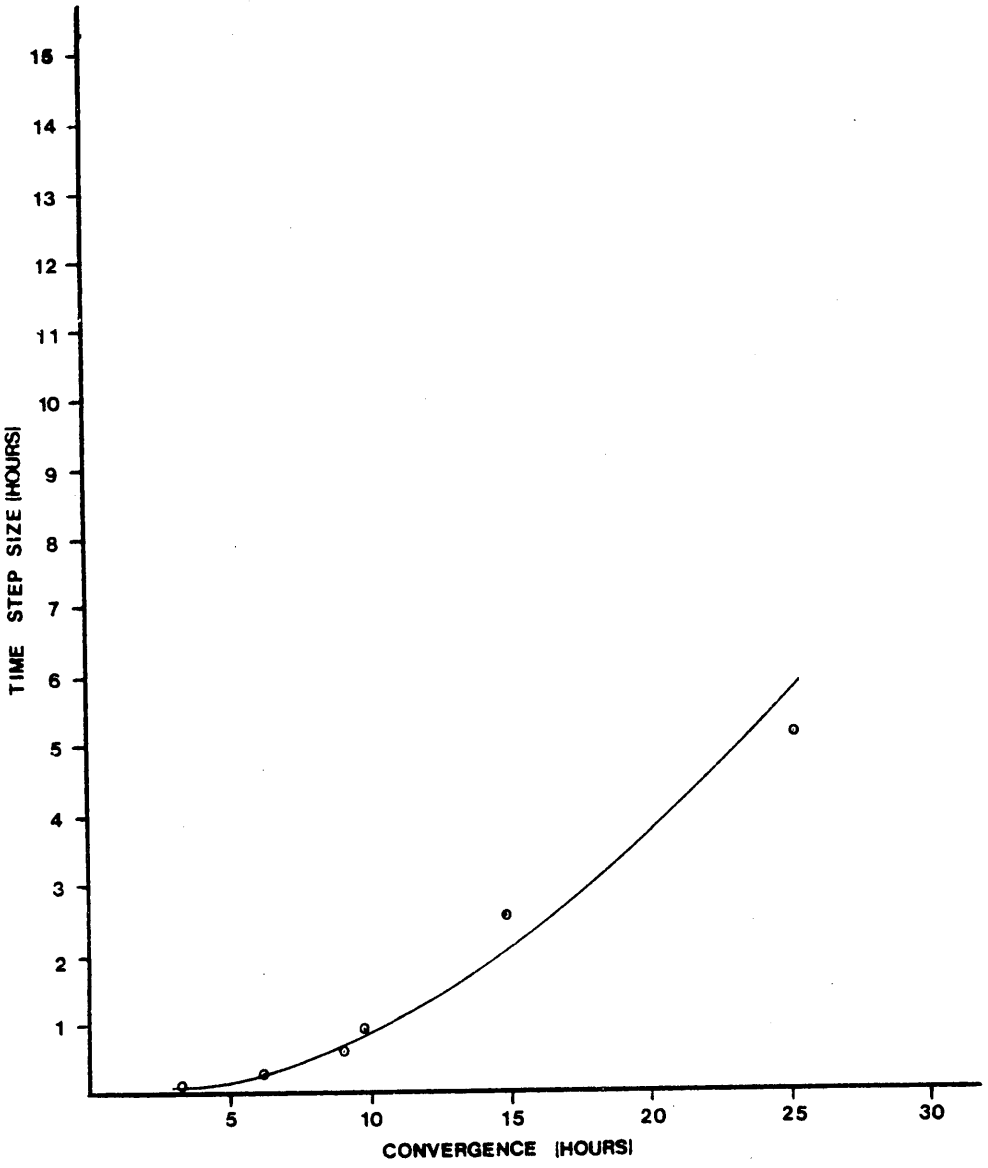
Parameters are:

NBAND = 26 (not necessarily exact, but must be ⩾ actual bandwidth
NNODES = 60 (number of nodes)
NELE = 3 (number of elements)
NBRICK = 3 (number of brick elements)
NTETRA = 0 (number of tetrahedron elements)
NUMBC = 9 (number of specified temperature nodes)
THETA = time increment, in hours
DAYS = simulation duration, in hours
XKX,XKY,XKZ = 31.4 (thermal conductivity, in Btu/hr.-ft.-°F
(Note that the brick element nodal sequence input is:)
ELEMENT 1: 1,10,18,19,20,12,3,2,11,4,13,21,22,23,14,6,5,7,15,24,25,26,17, 9,8,16.
ELEMENT 2: 18,27,35,36, etc.
ELEMENT 3: 35,44,52,53, etc.

TABLE 1

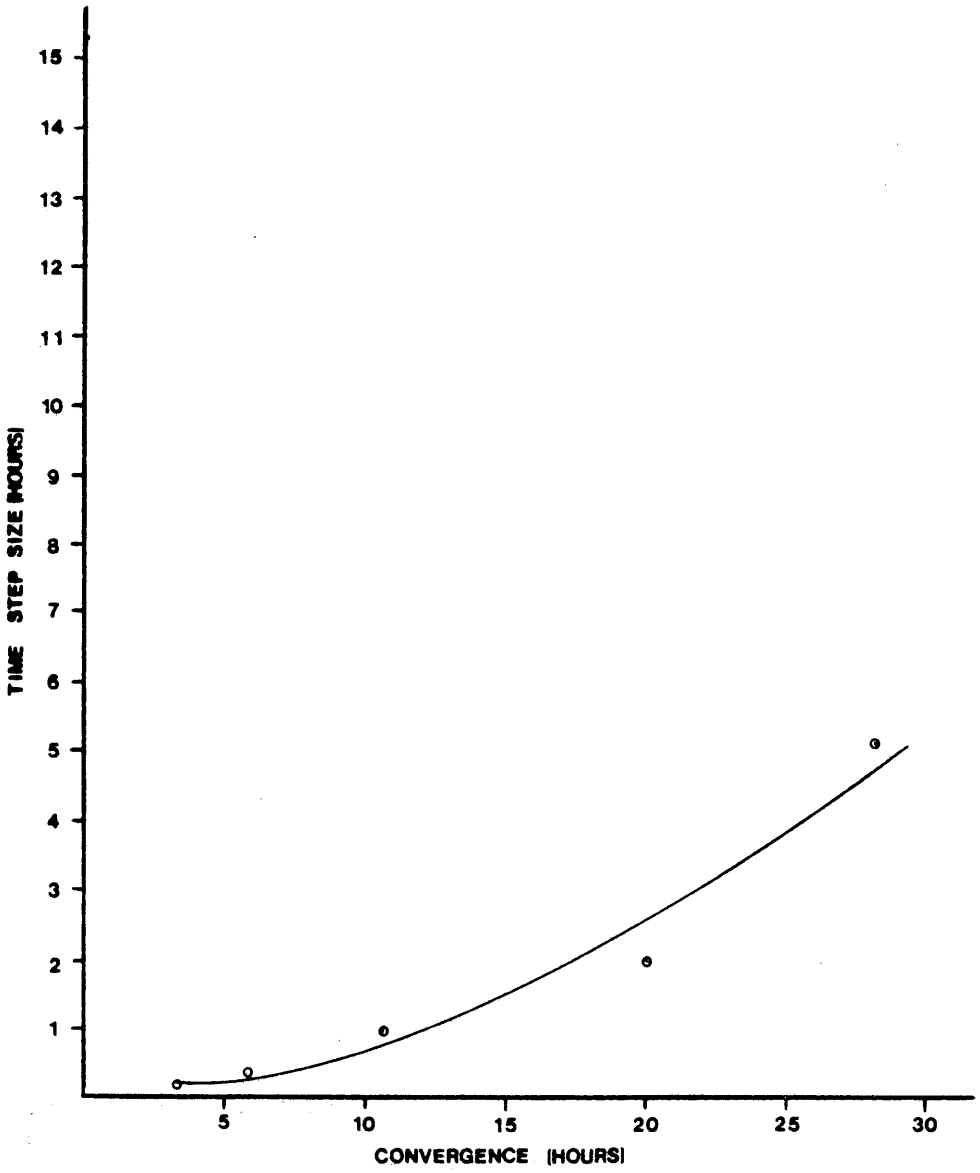
ANALYTICAL SOLUTION TO TEST PROBLEM

Time hours	Temperatures, °F, at Indicated Intervals		
	1 foot	2 foot	3 foot (midpoint)
1	40.01	92.29	99.25
2	-1.37	70.83	88.74
3	-13.19	50.36	73.62
4	-23.60	32.33	52.81
5	-32.75	16.47	34.49
6	-40.81	2.51	18.37
7	-47.91	-9.78	4.18
8	-54.15	-20.59	-8.31
9	-59.65	-30.11	-19.30
10	-64.48	-38.49	-28.97
15	-81.24	-67.51	-62.49
20	-90.09	-82.84	-80.19
24	-94.06	-89.70	-88.11



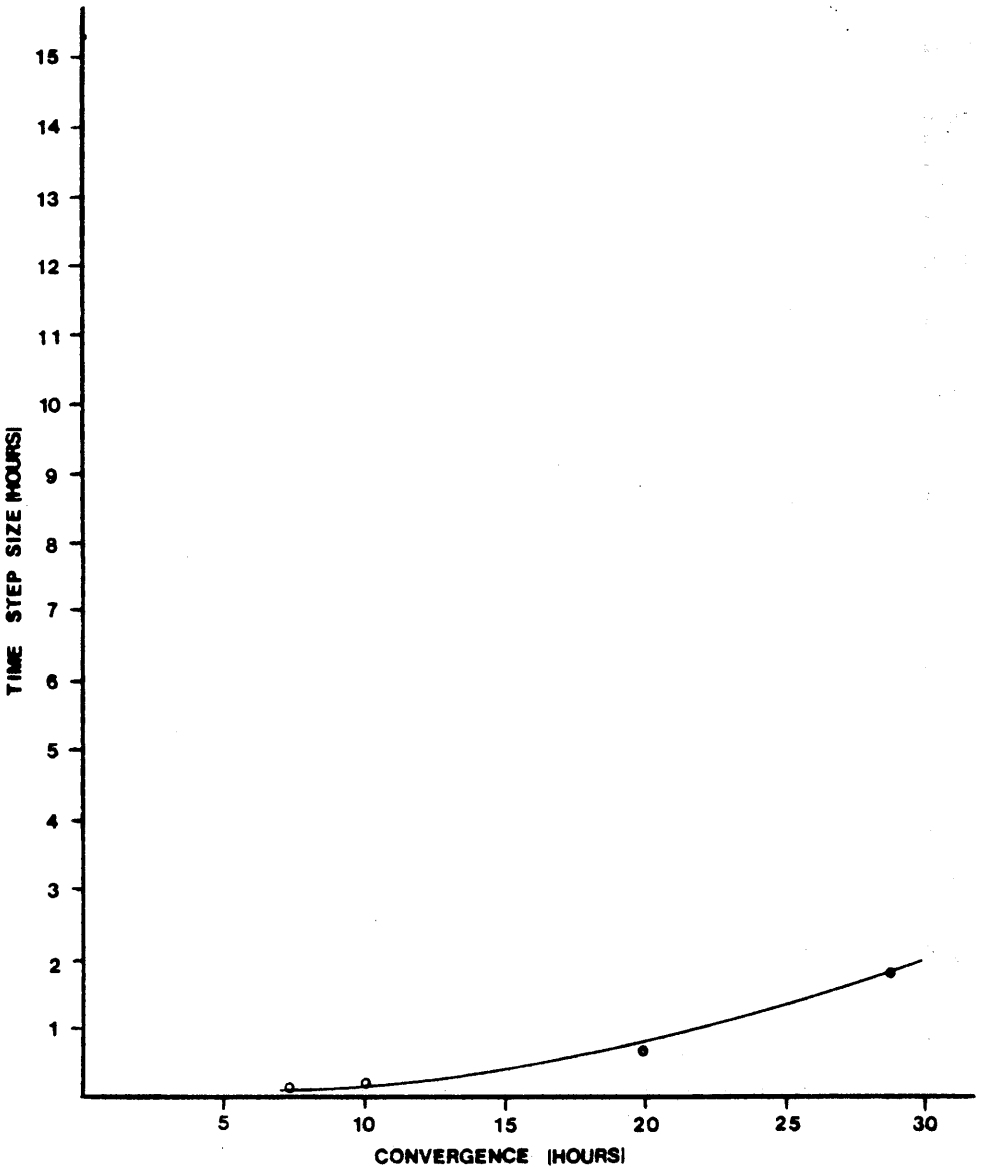
1 Foot Dimensions
2-D Problem
200° Shock

FIG. 11



1.5 Foot Dimensions
2-D Problem
200° Shock

FIG. 12



3 Foot Dimensions
2-D Problem
200° Shock

FIG. 13

REFERENCES

Bayfus, G.R. and G.L. Guymon, Thermal Analysis of Frozen Embankment Islands, Journal of Waterways, Harbors, and Coastal Engineering Division, ASCE. Vol 102, No. WW2, pp 123-139, 1976.

Bird, Stewart, and Lightfoot, Transport Phenomena, J. Wiley and Sons, 1960.

Desai and Abel, Introduction To The Finite Element Method, Van Nostraud Reinhold Co., 1972.

Huebner, K.H., The Finite Element Method For Engineers, J. Wiley and Sons, 1975.

Myers, G.E., Analytical Methods in Conduction Heat Transfer, McGraw-Hill, 1971.

Segerlind, L.J., Applied Finite Element Analysis, J. Wiley and Sons, 1976.

Zienkiewicz, O.C., The Finite Element Method In Engineering Science, McGraw-Hill, 1971.

APPENDIX A

THREE DIMENSIONAL FINITE ELEMENT MODEL DERIVATION

GOVERNING EQUATION

The governing partial differential equation for the transient heat conduction model in three dimensions is

$$k_x \frac{\partial^2 t}{\partial x^2} + k_y \frac{\partial^2 t}{\partial y^2} + k_z \frac{\partial^2 t}{\partial z^2} = \rho c \frac{\partial t}{\partial \theta} \tag{1}$$

where (a) kx, ky, kz are constant directinal thermal conductivities.

(b) θ is the time variable

(c) ρ is constant density

(d) c is the constant pressure specific heat

(e) an initial condition is given

(f) Boundary values are specified

(g) $\partial t/\partial n = 0$ along the surface of the volume being studied (i.e., the volume is thermally insulated).

Although Equation 1 assumes constant physical and thermal properties throughout the control volume, variable properties can be handled by the numerical methods proposed (see section entitled "Derivation of System Matrices,") contained in this appendix).

DISCRETIZATION

Given a volume V, discretize V into a finite union of "m" tetrahedra shaped elements $V^{(e)}$ where

$$V \approx \bigcup_{e=1}^{m} V^{(e)}$$

and the intersection of any two $V^{(e)}$ is an entire common face, edge or vertex. For each tetrahedra, specific nodal points at each vertex and at the midpoint of each edge. This will result in 10 nodes for each element $V^{(e)}$. Due to the discretization of a continuous volume, nearly all of the nodes will be shared by other elements. A special requirement of the discretization process is that

two contintiguous elements must share an entire face or an entire edge rather than portions or segments. This must result in 6 common nodes (face) or 3 common nodes (edge) for contiguous elements. Only when elements touch at a vertex will there be just one common node.

ELEMENT AND NODAL NUMBER

Number the elements and nodes of the descretized volume. Do not renumber shared nodes of contiguous elements. That is, if the system is discretized into two tetrahedron contiguous along a face, there will be 2 elements and 14 nodes in the model, not 2 elements and 20 nodes.

DERIVATION OF SYSTEM MATRICES

It was shown by Desai and Abel (1972) that solving Equation 1 along with its specifications is equivalent to minimizing with respect to the variable "t" (temperature) the variational statement

$$I = \frac{1}{2} \iiint_V \left(k_x \left(\frac{\partial t}{\partial x}\right)^2 + k_y \left(\frac{\partial t}{\partial y}\right)^2 + k_z \left(\frac{\partial t}{\partial z}\right)^2 + \rho c \left(\frac{\partial t^2}{\partial \theta}\right)\right) dV \qquad (2)$$

where we assume $\partial t/\partial n = 0$ along the volume's surface.

If we could minimize Equation 2 with respect to "t", we would arrive at an expression for temperature as a function of position and time. Instead of attempting to solve the above analytically, approximations will be used which when substituted into Equation 2 result in a system of linear equations with values of temperatures as the unknowns.

The first step is to formulate a function for temperature within the volume V. This will be done by specifying functions for temperature within each element $V^{(e)}$. These temperature functions will be continuous along the element interfaces such that Equation 2 will be defined.

Consider an element $V^{(e)}$ along with its ten specified nodal points. Let

$$t = f(t_1, t_2, \ldots, t_{10}, x, y, z)$$

where "t" is the temperature at point "P" having coordinates (x,y,z) of the volume V, and "t" is a function of the temperature of the element's nodal points where P is contained in $V^{(e)}$, $((t_1,t_2,\ldots,t_{10}))$ are the element's nodal temperatures in the order shown in Figure 5. This temperature function will equal the nodal temperature at each nodal point, and will interpolate between these nodal points to evaluate temperatures elsewhere within $V^{(e)}$.

Using the "local coordinate" system defined in Chapter 4, we can write such an interpolating temperature function as

$$t = (2L_1^2 - L_1)t_1 + (2L_2^2 - L_2)t_2 + (2L_3^2 - L_3)t_3$$
$$+ (2L_4^2 - L_4)t_4 + 4L\,L_2t_5 + 4L_1L_3t_6 + 4L_1L_4t_7$$
$$+ 4L_2L_3t_8 + 4L_3L_4t_9 + 4L_2L_4t_{10}$$

or in matrix notation

$$t = [(2L_1^2-L_1),\ (2L_2^2-L_2),\ 2L_3^2-L_3),\ (2L_4^2-L_4),\ 4L_1L_2, 4L_1L_3,\ 4L_1L_4,\ 4L_2L_3,\ 4L_3L_4,\ 4L_2L_4]\ \{t^{(e)}\} \qquad (3)$$

where $\underset{\sim}{t}^{(e)} = \{t_i\} = [t_1,t_2,\ldots,t_{10}]$

Equation 3 is valid for each element $V^{(e)}$ within the volume V. Also, this equation represents a quadratic variation of "t" (temperature) throughout each $V^{(e)}$, providing increased accuracy over a linear interpolating temperature function.

The second step is to evaluate the integral expression shown in Equation 2.

Assuming that

$$V \approx \bigcup_{e=1}^{m} V^{(e)}$$

and assuming Equation 2 can be rewritten as

$$I \approx \left(\frac{1}{2}\right) \sum_{e=1}^{m} \iiint_{V^{(e)}} \left(k_x \left(\frac{\partial t}{\partial x}\right)^2 + k_y \left(\frac{\partial t}{\partial y}\right)^2 k_z \left(\frac{\partial t}{\partial z}\right)^2 + \rho c \frac{\partial t^2}{\partial \theta}\right) dV \qquad (4)$$

then Equations 2 and 4 can be combined and expanded into

$$I \approx \sum_{e=1}^{m} \left(\frac{1}{2}\right) \iiint_{V^{(e)}} \left(k_x \left(\frac{\partial t}{\partial y}\right)^2 + k_y \left(\frac{\partial t^2}{\partial y}\right) + k_z \left(\frac{\partial t}{\partial z}\right)^2 + \rho c \frac{\partial t^2}{\partial \theta}\right) dV \tag{5}$$

Thus we can consider each element $V^{(e)}$ individually in evaluating 5, such that

$$I \approx \sum_{e=1}^{m} I^{(e)} \tag{6}$$

where

$$I^{(e)} = \frac{1}{2} \iiint_{V^{(e)}} \left(k_x \left(\frac{\partial t}{\partial x}\right)^2 + k_y \left(\frac{\partial t}{\partial y}\right)^2 + k_z \left(\frac{\partial t}{\partial z}\right)^2 + \rho c \frac{\partial t^2}{\partial \theta}\right) dV$$

The third step is to note that since we have an interpolating temperature function that is valid for each element $V^{(e)}$, we can approximate the partial derivatives in Equation 6 by the differential formulae established in Chapter 4. Furthermore, these expressions will be valid for each element.

To simplify calculations, rewrite the expression for $I^{(e)}$ in Equation 6 as

$$I^{(e)} = \frac{1}{2} \iiint_{V^{(e)}} k_x \left(\frac{\partial t}{\partial x}\right)^2 dV + \frac{1}{2} \iiint_{V^{(e)}} k_y \left(\frac{\partial t}{\partial y}\right)^2 dV + \frac{1}{2} \iiint_{V^{(e)}} k_z \left(\frac{\partial t}{\partial z}\right)^2 dV + \frac{1}{2} \iiint_{V^{(e)}} \rho c \frac{\partial t^2}{\partial \theta} dV \tag{7}$$

or simply

$$I^{(e)} = I_x^{(e)} + I_y^{(e)} + I_z^{(e)} + I_{\rho c}^{(e)} \tag{8}$$

where the subscripts refer to the identifying governing parameter for each expression in Equation 7. The reason for this is $I_x^{(3)}$, $I_y^{(e)}$ and $I_z^{(e)}$ are similar, except for some multiplying constants of position and thermal parameters. Hence, we only need to find a useable expression for $I_x^{(e)}$, and then expand the results to include $I_y^{(e)}$ and $I_z^{(e)}$. $I_{\rho c}^{(e)}$ must be solved separately due to the time derivative.

Our problem is to minimize Equation2 with respect to "t" (temperature). This is equivalent to minimizing Equation 4 with respect to temperature. But our interpolating temperature function is a function of nodal temperatures. Thus we must minimize Equation 5 with respect to each nodal temperature of the entire volume V. By Equation 6, we see that we must minimize each $I^{(e)}$ with respect to each nodal temperature of V. Since only ten nodes are contained in each $V^{(e)}$, most nodal temperature minimization efforts (for a several-element nodel) equate to zero. That is, minimizing Equation 4 with respect to each of the volume's V nodal temperatures is equivalent to the sum of the minimizations of $I^{(e)}$ with respect to each nodal temperature of $V^{(e)}$.

Hence, for each $V^{(e)}$ we want to evaluate

$$\frac{\partial I^{(e)}}{\partial t_1} = 0, \quad \frac{\partial I^{(e)}}{\partial t_2} = 0, \ldots, \quad \frac{\partial I^{(e)}}{\partial t_{10}} = 0$$

where t_i is the nodal temperature as expressed in Equation 3.

Since

$$I^{(e)} = I_x^{(e)} + I_y^{(e)} + I_z^{(e)} + I_{\rho c}^{(e)}$$

we can minimize $I_x^{(e)}$ with respect to $(t_1, t_2, \ldots, t_{10})$ of $V^{(e)}$, and then extend these results to the minimization of $I_y^{(e)}$ and $I_z^{(e)}$. However, $I_{\rho c}^{(e)}$ must be minimized with respect to $(t_1, t_2, \ldots, t_{10})$ of $V^{(e)}$ separately.

Step four is performing the above described operations. We will use the notations a_i, b_i and c_i for the cofactors of the element's coordinate matrices (see Chapter 4) and Equation 3 for the interpolating temperature function. Consider, for any $V^{(e)}$

$$I_x^{(e)} = \frac{1}{2} k_x \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right)^2 dV$$

$$\frac{\partial I_x^{(e)}}{\partial t_1} = k_x \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) \frac{\partial \left(\frac{\partial t}{\partial x}\right)}{\partial t_1} dV$$

$$= k_x \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) \frac{a_1}{6V^{(e)}} (4L_1 - 1)\, dV$$

$$= \frac{a_1 k_x}{(6V^{(e)})^2} \iiint_{V^{(e)}} \{a_1[(16L_1^2 - 8L_1 + 1)t_1 + 4(4L_1L_2 - L_2)t_5 + 4(4L_1L_3 - L_3)t_6 + 4(4L_1L_4 - L_4)t_7]$$
$$+ a_2[(16L_1L_2 - 4L_1 - 4L_2 + 1)t_2 + 4(4L_1^2 - L_1)t_5 + 4(4L_1L_3 - L_3)t_8 + 4(4L_1L_4 - L_4)t_{10}]$$
$$+ a_3[(16L_1L_3 - 4L_1 - 4L_3 + 1)t_3 + 4(4L_1^2 - L_1)t_6 + 4(4L_1L_2 - L_2)t_8 + 4(4L_1L_4 - L_4)t_9]$$
$$+ a_4[(16L_1L_4 - 4L_1 - 4L_4 + 1)t_4 + 4(4L_1^2 - L_1)t_7 + 4(4L_1L_3 - L_3)t_9 + 4(4L_1L_2 - L_2)t_{10}]\} dV$$

where the integration is solved as follows:

i) $\iiint_{V^{(e)}} L_i^2\, dV = (6V^{(e)})\left(\frac{2}{120}\right)$

ii) $\iiint_{V^{(e)}} L_i\, dV = (6V^{(e)})\left(\frac{1}{24}\right)$

iii) $\iiint_{V^{(e)}} L_i L_j\, dV = (6V^{(e)})\left(\frac{1}{120}\right)$

iv) $\iiint_{V^{(e)}} dV = (6V^{(e)})\left(\frac{1}{6}\right)$

where,

$$\iiint_{V^{(e)}} L_1^p L_2^q L_3^r L_4^s\, dV = [(p!q!r!s!)/(p+q+r+s+3)!](6V^{(e)})$$

$$\frac{\partial I_x^{(e)}}{\partial t_1} = \frac{a_1 k_x}{6V^{(e)}} \{a_1\left[\tfrac{1}{10}t_1 + \tfrac{-1}{30}t_5 + \tfrac{-1}{30}t_6 + \tfrac{-1}{30}t_7\right]$$
$$+ a_2\left[\tfrac{-1}{30}t_2 + \tfrac{1}{10}t_5 + \tfrac{-1}{30}t_8 + \tfrac{-1}{30}t_{10}\right]$$
$$+ a_3\left[\tfrac{-1}{30}t_3 + \tfrac{1}{10}t_6 + \tfrac{-1}{30}t_8 + \tfrac{-1}{30}t_9\right]$$
$$+ a_4\left[\tfrac{-1}{30}t_4 + \tfrac{1}{10}t_7 + \tfrac{-1}{30}t_9 + \tfrac{-1}{30}t_{10}\right]\}$$

The other partial derivatives follow:

$$\frac{\partial I_x^{(e)}}{\partial t_2} = k_x \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) \frac{a_2}{6V^{(e)}} (4L_2 - 1)\, dV$$

$$= \frac{a_2 k_x}{6V^{(e)}} \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right)(4L_2 - 1)\, dV$$

$$= \frac{a_2 k_x}{6V^{(e)}} \left\{ a_1 \left[\frac{-1}{30} t_1 + \frac{1}{10} t_5 + \frac{-1}{30} t_6 + \frac{-1}{30} t_7\right] \right.$$

$$+ a_2 \left[\frac{1}{10} t_2 + \frac{-1}{30} t_5 + \frac{-1}{30} t_8 + \frac{-1}{30} t_{10}\right]$$

$$+ a_3 \left[\frac{-1}{30} t_3 + \frac{-1}{30} t_6 + \frac{1}{10} t_8 + \frac{-1}{30} t_9\right]$$

$$\left. + a_4 \left[\frac{-1}{30} t_4 + \frac{-1}{30} t_7 + \frac{-1}{30} t_9 + \frac{1}{10} t_{10}\right] \right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_3} = k_x \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) \frac{a_3}{6V^{(e)}} (4L_3 - 1)\, dV$$

$$= \frac{a_3 k_x}{6V^{(e)}} \left\{ a_1 \left[\frac{-1}{30} t_1 + \frac{-1}{30} t_5 + \frac{1}{10} t_6 + \frac{-1}{30} t_7\right] \right.$$

$$+ a_2 \left[\frac{-1}{30} t_2 + \frac{-1}{30} t_5 + \frac{1}{10} t_8 + \frac{-1}{30} t_{10}\right]$$

$$+ a_3 \left[\frac{1}{10} t_3 + \frac{-1}{30} t_6 + \frac{-1}{30} t_8 + \frac{-1}{30} t_9\right]$$

$$\left. + a_4 \left[\frac{-1}{30} t_4 + \frac{-1}{30} t_7 + \frac{1}{10} t_9 + \frac{-1}{30} t_{10}\right] \right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_4} = k_x \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) \frac{a_4}{6V^{(e)}} (4L_4 - 1)\, dV$$

$$= \frac{a_4 k_x}{6V^{(e)}} \left\{ a_1 \left[\frac{-1}{30} t_1 + \frac{-1}{30} t_5 + \frac{-1}{30} t_6 + \frac{1}{10} t_7\right] \right.$$

$$+ a_2 \left[\frac{-1}{30} t_2 + \frac{-1}{30} t_5 + \frac{-1}{30} t_8 + \frac{1}{10} t_{10}\right]$$

$$+ a_3 \left[\frac{-1}{30} t_3 + \frac{-1}{30} t_6 + \frac{-1}{30} t_8 + \frac{1}{10} t_9\right]$$

$$\left. + a_4 \left[\frac{1}{10} t_4 + \frac{-1}{10} t_7 + \frac{-1}{30} t_9 + \frac{-1}{30} t_{10}\right] \right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_5} = k_x \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) \frac{4}{6V^{(e)}} (a_1 L_2 + a_2 L_1)\, dV$$

$$= \frac{4a_1 k_x}{6V^{(e)}} \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) L_2\, dV + \frac{4a_2 k_x}{6V^{(e)}} \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) L_1\, dV$$

$$= \frac{4a_1 k_x}{6V^{(e)}} \left\{ a_1 \left[\frac{-1}{120} t_1 + \frac{1}{15} t_5 + \frac{1}{30} t_6 + \frac{1}{30} t_7\right] \right.$$
$$+ a_2 \left[\frac{1}{40} t_2 + \frac{1}{30} t_5 + \frac{1}{30} t_8 + \frac{1}{30} t_{10}\right]$$
$$+ a_3 \left[\frac{-1}{120} t_3 + \frac{1}{30} t_6 + \frac{1}{15} t_8 + \frac{1}{30} t_9\right]$$
$$\left. + a_4 \left[\frac{-1}{120} t_4 + \frac{1}{30} t_7 + \frac{1}{30} t_9 + \frac{1}{15} t_{10}\right] \right\}$$
$$+ \frac{4a_2 k_x}{6V^{(e)}} \left\{ a_1 \left[\frac{1}{40} t_1 + \frac{1}{30} t_5 + \frac{1}{30} t_6 + \frac{1}{30} t_7\right] \right.$$
$$+ a_2 \left[\frac{-1}{120} t_2 + \frac{1}{15} t_5 + \frac{1}{30} t_8 + \frac{1}{30} t_{10}\right]$$
$$+ a_3 \left[\frac{-1}{120} t_3 + \frac{1}{15} t_6 + \frac{1}{30} t_8 + \frac{1}{30} t_9\right]$$
$$\left. + a_4 \left[\frac{-1}{120} t_4 + \frac{1}{15} t_7 + \frac{1}{30} t_9 + \frac{1}{30} t_{10}\right] \right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_6} = \frac{4}{6} \frac{K_x}{V^{(e)}} \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right) (a_1 L_3 + a_3 L_1)\, dV$$

$$= \frac{4k_x a_1}{6V^{(e)}} \left\{ a_1 \left[\frac{-1}{120} t_1 + \frac{1}{30} t_5 + \frac{1}{15} t_6 + \frac{1}{30} t_7\right] \right.$$
$$+ a_2 \left[\frac{-1}{120} t_2 + \frac{1}{30} t_5 + \frac{1}{15} t_8 + \frac{1}{30} t_{10}\right]$$
$$+ a_3 \left[\frac{-1}{120} t_3 + \frac{1}{30} t_6 + \frac{1}{30} t_8 + \frac{1}{30} t_9\right]$$
$$\left. + a_4 \left[\frac{1}{40} t_4 + \frac{1}{30} t_7 + \frac{1}{15} t_9 + \frac{1}{30} t_{10}\right] \right\}$$
$$+ \frac{4k_x a_3}{6V^{(e)}} \left\{ a_1 \left[\frac{1}{40} t_1 + \frac{1}{30} t_5 + \frac{1}{30} t_6 + \frac{1}{30} t_9\right] \right.$$
$$+ a_2 \left[\frac{-1}{120} t_2 + \frac{1}{15} t_5 + \frac{1}{30} t_8 + \frac{1}{30} t_{10}\right]$$
$$+ a \left[\frac{-1}{120} t_4 + \frac{1}{15} t_7 + \frac{1}{30} t_9 + \frac{1}{30} t_{10}\right]$$
$$\left. + a_3 \left[\frac{-1}{120} t_3 + \frac{1}{15} t_6 + \frac{1}{30} t_8 + \frac{1}{30} t_9\right] \right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_7} = \frac{4k_x}{6V^{(e)}} \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right)(a_1 L_4 + a_4 L_1)\, dV$$

$$= \frac{4k_x a_1}{6V^{(e)}}\left\{ a_1\left[\frac{-1}{120}t_1 + \frac{1}{30}t_5 + \frac{1}{30}t_6 + \frac{1}{15}t_7\right] + a_2\left[\frac{-1}{120}t_2 + \frac{1}{30}t_5 + \frac{1}{30}t_8 + \frac{1}{15}t_{10}\right] + a_3\left[\frac{-1}{120}t_3 + \frac{1}{30}t_6 + \frac{1}{30}t_8 + \frac{1}{15}t_9\right] + a_4\left[\frac{1}{40}t_4 + \frac{1}{30}t_7 + \frac{1}{30}t_9 + \frac{1}{30}t_{10}\right]\right\}$$
$$+ \frac{4k_x a_4}{6V^{(e)}}\left\{ a_1\left[\frac{1}{40}t_1 + \frac{1}{30}t_5 + \frac{1}{30}t_6 + \frac{1}{30}t_7\right] + a_2\left[\frac{-1}{120}t_2 + \frac{1}{15}t_5 + \frac{1}{30}t_8 + \frac{1}{30}t_{10}\right] + a_3\left[\frac{-1}{120}t_3 + \frac{1}{15}t_6 + \frac{1}{30}t_8 + \frac{1}{30}t_9\right] + a_4\left[\frac{-1}{120}t_4 + \frac{1}{15}t_7 + \frac{1}{30}t_9 + \frac{1}{30}t_{10}\right]\right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_8} = \frac{4k_x}{6V^{(e)}} \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right)(a_2 L_3 + a_3 L_2)\, dV$$

$$= \frac{4k_x a_2}{6V^{(e)}}\left\{ a_1\left[\frac{-1}{120}t_1 + \frac{1}{30}t_5 + \frac{1}{15}t_6 + \frac{1}{30}t_7\right] + a_2\left[\frac{-1}{120}t_2 + \frac{1}{30}t_5 + \frac{1}{15}t_8 + \frac{1}{30}t_{10}\right] + a_3\left[\frac{1}{40}t_3 + \frac{1}{30}t_6 + \frac{1}{30}t_8 + \frac{1}{30}t_9\right] + a_4\left[\frac{-1}{120}t_4 + \frac{1}{30}t_7 + \frac{1}{15}t_9 + \frac{1}{30}t_{10}\right]\right\}$$
$$+ \frac{4k_x a_3}{6V^{(e)}}\left\{ a_1\left[\frac{-1}{120}t_1 + \frac{1}{15}t_5 + \frac{1}{30}t_6 + \frac{1}{30}t_7\right] + a_2\left[\frac{1}{40}t_2 + \frac{1}{30}t_5 + \frac{1}{30}t_8 + \frac{1}{30}t_{10}\right] + a_3\left[\frac{-1}{120}t_3 + \frac{1}{30}t_6 + \frac{1}{15}t_8 + \frac{1}{30}t_9\right] + a_4\left[\frac{-1}{120}t_4 + \frac{1}{30}t_7 + \frac{1}{30}t_9 + \frac{1}{15}t_{10}\right]\right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_9} = \frac{4k_x}{6V^{(e)}} \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right)(a_3L_4 + a_4L_3)\,dV$$

$$= \frac{4k_xa_3}{6V^{(e)}}\left\{a_1\left[\frac{-1}{120}t_1 + \frac{1}{30}t_5 + \frac{1}{30}t_6 + \frac{1}{15}t_7\right]\right.$$
$$+ a_2\left[\frac{-1}{120}t_2 + \frac{1}{30}t_5 + \frac{1}{30}t_8 + \frac{1}{15}t_{10}\right]$$
$$+ a_3\left[\frac{-1}{120}t_3 + \frac{1}{30}t_6 + \frac{1}{30}t_8 + \frac{1}{15}t_9\right]$$
$$\left.+ a_4\left[\frac{1}{40}t_4 + \frac{1}{30}t_7 + \frac{1}{30}t_9 + \frac{1}{30}t_{10}\right]\right\}$$
$$+ \frac{4k_xa_4}{6V^{(e)}}\left\{a_1\left[\frac{-1}{120}t_1 + \frac{1}{30}t_5 + \frac{1}{15}t_6 + \frac{1}{30}t_7\right]\right.$$
$$+ a_2\left[\frac{-1}{120}t_2 + \frac{1}{30}t_5 + \frac{1}{15}t_8 + \frac{1}{30}t_{10}\right]$$
$$+ a_3\left[\frac{1}{40}t_3 + \frac{1}{30}t_6 + \frac{1}{30}t_8 + \frac{1}{30}t_9\right]$$
$$\left.+ a_4\left[\frac{-1}{120}t_4 + \frac{1}{30}t_7 + \frac{1}{15}t_9 + \frac{1}{30}t_{10}\right]\right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_{10}} = \frac{4k_x}{6V^{(e)}} \iiint_{V^{(e)}} \left(\frac{\partial t}{\partial x}\right)(a_2L_4 + a_4L_2)\,dV$$

$$= \frac{4k_xa_2}{6V^{(e)}}\left\{a_1\left[\frac{-1}{120}t_1 + \frac{1}{30}t_5 + \frac{1}{30}t_6 + \frac{1}{15}t_7\right]\right.$$
$$+ a_2\left[\frac{-1}{120}t_2 + \frac{1}{30}t_5 + \frac{1}{30}t_8 + \frac{1}{15}t_{10}\right]$$
$$+ a_3\left[\frac{-1}{120}t_3 + \frac{1}{30}t_6 + \frac{1}{30}t_8 + \frac{1}{15}t_9\right]$$
$$\left.+ a_4\left[\frac{1}{40}t_4 + \frac{1}{30}t_7 + \frac{1}{30}t_9 + \frac{1}{30}t_{10}\right]\right\}$$
$$+ \frac{4k_xa_4}{6V^{(e)}}\left\{a_1\left[\frac{-1}{120}t_1 + \frac{1}{15}t_5 + \frac{1}{30}t_6 + \frac{1}{30}t_7\right]\right.$$
$$+ a_2\left[\frac{1}{40}t_2 + \frac{1}{30}t_5 + \frac{1}{30}t_8 + \frac{1}{30}t_{10}\right]$$
$$+ a_3\left[\frac{-1}{120}t_3 + \frac{1}{30}t_6 + \frac{1}{15}t_8 + \frac{1}{30}t_9\right]$$
$$\left.+ a_4\left[\frac{-1}{120}t_4 + \frac{1}{30}t_7 + \frac{1}{30}t_9 + \frac{1}{15}t_{10}\right]\right\}$$

The preceeding processes of partial differentiation must be repeated for $I_y^{(e)}$ and $I_z^{(e)}$. However, parallel development will result in the same expressions as derived for $I_x^{(e)}$, except that for $I_y^{(e)}$ we must substitute "b_i" for "a_i" , and "k_y" for k_x ; and for $I_z^{(e)}$ we must substitute "c_i" for "a_i", and "k_z" for "k_x".

Let $\underset{\sim}{t}^{(e)}$ = the column vector $[t_1, t_2, \ldots, t_{10}]^T$. Then minimizing $I_x^{(e)} + I_y^{(e)} + I_z^{(e)}$ with respect to $(t_1, t_2, \ldots, t_{10})$ of $V^{(e)}$ is equivalent to

$$\frac{\partial(I_x^{(e)} + I_y^{(e)} + I_z^{(e)})}{\partial \underset{\sim}{t}^{(e)}} = 0 \tag{9}$$

By the above derivations, we see that Equation 9 establishes the matrix expression

$$[X\ K]\ \underset{\sim}{t}^{(e)} = 0 \tag{10}$$

where [X K] is the "element conduction matrix" for $V^{(e)}$ where $\underset{\sim}{t}^{(e)}$ is the column vector of nodal temperatures of $V^{(e)}$.

Note that we can expand Equation 10 as

$$[X\ K]\ \underset{\sim}{t}^{(e)} = \{[K_x^{(e)}] + [K_y^{(e)}] + [K_z^{(e)}]\}\ \underset{\sim}{t}^{(e)} = 0$$

where each matrix in the above sum corresponds to the minimization of its respective entry in Equation 9. The matrix $[K_x^{(e)}]$ is written in modified form in Figure A-1 of the appendix. The matrices $[K_y^{(e)}]$ and $[K_z^{(e)}]$ can be found by substituting the constants b_i for the a_i and k_y for k_x (for matrix $[K_y^{(e)}]$); and by substituting c_i for the a_i and k_z for k_x (for matrix $[K_z^{(e)}]$.

Equation 10 only represents part of the expression for minimizing

Equation 7. We must still minimize $I_{\rho c}^{(e)}$ with respect to $(t_1, t_2, \cdots, t_{10})$ of $V^{(e)}$, that is we must solve

$$\frac{\partial I_{\rho c}^{(e)}}{\partial \underset{\sim}{t}^{(e)}} = 0$$

Consider for any $V^{(e)}$, $I_{\rho c}^{(e)} = \frac{1}{2} \iiint_{V^{(e)}} (\rho c)^{(e)} \frac{\partial t^2}{\partial \theta} \, dV$; θ = time,

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_1} = (\rho c)^{(e)} \frac{d}{d\theta} \iiint_{V^{(e)}} [N] \underset{\sim}{t}^{(e)} (2L_1^2 - L_1) \, dV$$

$$= (\rho c)^{(e)} \iiint_{V^{(e)}} \begin{bmatrix} (4L_1^4 - 4L_1^3 + L_1^2) \\ (4L_2^2 L_1^2 - 2L_2^2 L_1 - 2L_2 L_1^2 + L_1 L_2) \\ (4L_3^2 L_1^2 - 2L_3^2 L_1 - 2L_3 L_1^2 + L_1 L_3) \\ (4L_4^2 L_1^2 - 2L_4^2 L_1 - 2L_4 L_1^2 + L_1 L_4) \\ (8L_1^3 L_2 - 4L_1^2 L_2) \\ (8L_1^3 L_3 - 4L_1^2 L_3) \\ (8L_1^3 L_4 - 4L_1^2 L_4) \\ (8L_1^2 L_2 L_3 - 4L_1 L_2 L_3) \\ (8L_1^2 L_3 L_4 - 4L_1 L_3 L_4) \\ (8L_1^2 L_2 L_4 - 4L_1 L_2 L_4) \end{bmatrix}^T \frac{d \underset{\sim}{t}^{(e)}}{d\theta} \, dV$$

$$= (\rho c)^{(e)} (6V^{(e)}) \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_2 \\ \alpha_2 \\ \alpha_3 \\ \alpha_3 \\ \alpha_3 \\ \alpha_4 \\ \alpha_4 \\ \alpha_4 \end{bmatrix}^T \dot{\underset{\sim}{t}}^{(e)}$$

$$\alpha_1 = \left(4 \cdot \frac{4!}{7!} - 4 \cdot \frac{3!}{6!} + \frac{2}{5!}\right)$$
$$\alpha_2 = \left(4 \cdot \frac{4}{7!} - 4 \cdot \frac{2}{6!} + \frac{1}{5!}\right)$$
$$\alpha_3 = \left(8 \cdot \frac{3!}{7!} - 4 \cdot \frac{2}{6!}\right)$$
$$\alpha_4 = \left(8 \cdot \frac{2}{7!} - 4 \cdot \frac{1}{6!}\right)$$

$$\frac{\partial I_{pc}^{(e)}}{\partial t_5} = \frac{1}{2}(\rho c)^{(e)} \frac{d}{d\theta} \iiint_{V^{(e)}} 2\,(t^{(e)})\, \frac{\partial t^{(e)}}{\partial t_5^{(e)}}\, dV$$

$$= (\rho c)^{(e)} \frac{d}{d\theta} \iiint_{V^{(e)}} [N]\, \underset{\sim}{t}^{(e)} (4L_1L_2)\, dV$$

$$= 4(\rho c)^{(e)} \iiint_{V^{(e)}} \begin{bmatrix} 2L_1^3L_2 - L_1^2L_2 \\ 2L_1L_2^3 - L_1L_2^2 \\ 2L_3^2L_1L_2 - L_3L_1L_2 \\ 2L_4^2L_1L_2 - L_4L_1L_2 \\ 4L_1^2L_2^2 \\ 4L_1^2L_3L_2 \\ 4L_1^2L_2L_4 \\ 4L_2^2L_1L_3 \\ 4L_1L_2L_3L_4 \\ 4L_2^2L_1L_4 \end{bmatrix} \frac{d\,\underset{\sim}{t}^{(e)}}{d\theta}\, dV$$

$$= 6(\rho cV)^{(e)} \begin{bmatrix} \beta_1 \\ \beta_1 \\ \beta_2 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_4 \\ \beta_4 \\ \beta_5 \\ \beta_4 \end{bmatrix} \underset{\sim}{\dot{t}}^{(e)}, \quad \underset{\sim}{\dot{t}}^{(e)} = \frac{d\,\underset{\sim}{t}^{(e)}}{d\theta}$$

$$\beta_1 = \left(2\cdot\frac{3!}{7!} - \frac{2}{6!}\right)4$$
$$\beta_2 = \left(2\cdot\frac{2}{7!} - \frac{1}{6!}\right)4$$
$$\beta_3 = \left(4\cdot\frac{4}{7!}\right)4$$
$$\beta_4 = \left(4\cdot\frac{2}{7!}\right)4$$
$$\beta_5 = \left(4\cdot\frac{1}{7!}\right)4$$

Similarly,

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_2} = 6(\rho c V)^{(e)} \left[\alpha_2\ \alpha_1\ \alpha_2\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_4\ \alpha_3\ \alpha_4\ \alpha_3\right] \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_3} = 6(\rho c V)^{(e)} \left[\alpha_2\ \alpha_2\ \alpha_1\ \alpha_2\ \alpha_4\ \alpha_3\ \alpha_4\ \alpha_3\ \alpha_3\ \alpha_4\right] \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_4} = 6(\rho c V)^{(e)} \left[\alpha_2\ \alpha_2\ \alpha_2\ \alpha_1\ \alpha_4\ \alpha_4\ \alpha_3\ \alpha_4\ \alpha_3\ \alpha_3\right] \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_5} = 6(\rho c V)^{(e)} \left[\beta_1\ \beta_1\ \beta_2\ \beta_2\ \beta_3\ \beta_4\ \beta_4\ \beta_4\ \beta_5\ \beta_4\right] \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_6} = 6(\rho c V)^{(e)} \left[\beta_1\ \beta_2\ \beta_1\ \beta_2\ \beta_4\ \beta_3\ \beta_4\ \beta_4\ \beta_4\ \beta_5\right] \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_7} = 6(\rho c V)^{(e)} \left[\beta_1\ \beta_2\ \beta_2\ \beta_1\ \beta_4\ \beta_4\ \beta_3\ \beta_5\ \beta_4\ \beta_4\right] \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_8} = 6(\rho c V)^{(e)} \left[\beta_2\ \beta_1\ \beta_1\ \beta_2\ \beta_4\ \beta_4\ \beta_5\ \beta_3\ \beta_4\ \beta_4\right] \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_9} = 6(\rho c V)^{(e)} \left[\beta_2\ \beta_2\ \beta_1\ \beta_1\ \beta_5\ \beta_4\ \beta_4\ \beta_4\ \beta_3\ \beta_4\right] \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_{10}} = 6(\rho c V)^{(e)} \left[\beta_2\ \beta_1\ \beta_2\ \beta_1\ \beta_4\ \beta_5\ \beta_4\ \beta_4\ \beta_4\ \beta_3\right] \dot{\underset{\sim}{t}}^{(e)}$$

Or: (where by symmetry $\beta_1 = \alpha_3$, $\beta_2 = \alpha_4$)

$$\frac{\partial I_{pc}^{(e)}}{\partial \underset{\sim}{t}^{(e)}} = G(pcV)^{(e)} \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_2 & \alpha_2 & \alpha_3 & \alpha_3 & \alpha_3 & \alpha_4 & \alpha_4 & \alpha_4 \\ \alpha_2 & \alpha_1 & \alpha_2 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_4 & \alpha_3 & \alpha_4 & \alpha_3 \\ \alpha_2 & \alpha_2 & \alpha_1 & \alpha_2 & \alpha_4 & \alpha_3 & \alpha_4 & \alpha_3 & \alpha_3 & \alpha_4 \\ \alpha_2 & \alpha_2 & \alpha_2 & \alpha_1 & \alpha_4 & \alpha_4 & \alpha_3 & \alpha_4 & \alpha_3 & \alpha_3 \\ \beta_1 & \beta_1 & \beta_2 & \beta_2 & \beta_3 & \beta_4 & \beta_4 & \beta_4 & \beta_5 & \beta_4 \\ \beta_1 & \beta_2 & \beta_1 & \beta_2 & \beta_4 & \beta_3 & \beta_4 & \beta_4 & \beta_4 & \beta_5 \\ \beta_1 & \beta_2 & \beta_2 & \beta_1 & \beta_4 & \beta_4 & \beta_3 & \beta_5 & \beta_4 & \beta_4 \\ \beta_2 & \beta_1 & \beta_1 & \beta_2 & \beta_4 & \beta_4 & \beta_5 & \beta_3 & \beta_4 & \beta_4 \\ \beta_2 & \beta_2 & \beta_1 & \beta_1 & \beta_5 & \beta_4 & \beta_4 & \beta_4 & \beta_3 & \beta_4 \\ \beta_2 & \beta_1 & \beta_2 & \beta_1 & \beta_4 & \beta_5 & \beta_4 & \beta_4 & \beta_4 & \beta_3 \end{bmatrix} \underset{\sim}{\dot{t}}^{(e)}$$

Minimizing $I_{pc}^{(e)}$ with respect to $(t_1, t_2, \cdots, t_{10})$ of $V^{(e)}$ results in the "element capacitance matrix" expression

$$[XC^{(e)}]\underset{\sim}{\dot{t}}^{(e)} = 0 \qquad (11)$$

What we wanted to do was minimize Equation 7 with respect to $(t_1, t_2, \cdots, t_{10})$ of $V^{(e)}$. Hence, combining Equations 7, 10, and 11 we get

$$[XK^{(e)}]\underset{\sim}{t}^{(e)} + [XC^{(e)}]\underset{\sim}{\dot{t}}^{(e)} = 0 \qquad (12)$$

By Equations 5 and 6, we must perform the above operations for each of the "m" elements, and sum the results in matrix form. This

combination of element matrices will form the "global matrix" system

$$[G\ K]\underset{\sim}{t} + [G\ C]\dot{\underset{\sim}{t}} = 0 \tag{13}$$

where $[G\ K]$ = the global conduction matrix

$[GC]$ = the global capacitance matrix

$\underset{\sim}{t}$ = the column vector of all nodal temperatures

$\dot{\underset{\sim}{t}}$ = the column vector of time derivatives of all nodal temperatures

Using Equation 13, we can apply a time advancement modification such that the "Crank-Nicolson" method, and formulate a matrix system that can be programmed into the computer. From the above, it can be seen that variable physical and thermal parameters can be handled by specifying constant parameters per each element $V^{(e)}$.

$$\frac{K_x}{180V^{(e)}} \begin{bmatrix} 3a_1^2 & -a_1a_2 & -a_1a_3 & -a_1a_4 & (3a_1a_2 - a_1^2) \\ -a_2a_1 & 3a_2^2 & -a_2a_3 & -a_2a_4 & (3a_2a_1 - a_2^2) \\ -a_3a_1 & -a_3a_2 & 3a_3^2 & -a_3a_4 & (-a_3a_1 - a_3a_2) \\ -a_4a_1 & -a_4a_2 & -a_4a_3 & 3a_4^2 & (-a_4a_1 - a_4a_2) \\ (3a_2a_1 - a_1^2) & (3a_1a_2 - a_2^2) & (-a_1a_3 - a_2a_3) & (-a_1a_4 - a_2a_4) & (8a_1^2 + 8a_1a_2 + 8a_2^2) \\ (3a_3a_1 - a_1^2) & (-a_1a_2 - a_3a_2) & (3a_1a_3 - a_3^2) & (-a_1a_4 - a_3a_4) & (4a_1^2 + 4a_1a_2 + 4a_4a_1 + 8a_4a_2) \\ (3a_4a_1 - a_1^2) & (-a_1a_2 - a_4a_2) & (-a_1a_3 - a_4a_3) & (3a_1a_4 - a_4^2) & (4a_1^2 + 4a_1a_2 + 4a_4a_1 + 8a_4a_2) \\ (-a_2a_1 - a_3a_1) & (3a_3a_2 - a_2^2) & (3a_2a_3 - a_3^2) & (-a_2a_4 - a_3a_4) & (4a_2a_1 + 4a_2^2 + 8a_3a_1 + 4a_3a_2) \\ (-a_3a_1 - a_4a_1) & (-a_3a_2 - a_4a_2) & (3a_4a_3 - a_3^2) & (3a_3a_4 - a_4^2) & (4a_3a_1 + 4a_3a_2 + 4a_1a_4 + 4a_4a_2) \\ (-a_2a_1 - a_4a_1) & (3a_4a_2 - a_2^2) & (-a_2a_3 - a_4a_3) & (3a_2a_4 - a_4^2) & (4a_2a_1 + 4a_2^2 + 8a_4a_1 + 4a_4a_2) \end{bmatrix}$$

COLUMNS 1-5 OF [X]

FIGURE A-1 (appendix A)

COLUMNS 6-10 OF [X|S]

$(3a_1a_3-a_1^2)$	$(3a_1a_4-a_1^2)$	$(-a_1a_2-a_1a_3)$	$(-a_1a_3-a_1a_4)$	$(-a_1a_2-a_1a_4)$
$(-a_2a_1-a_2a_3)$	$(-a_2a_1-a_2a_4)$	$(3a_2a_3-a_2^2)$	$(-a_2a_3-a_2a_4)$	$(3a_2a_4-a_2^2)$
$(3a_3a_1-a_3^2)$	$(-a_3a_1-a_3a_4)$	$(3a_3a_2-a_3^2)$	$(3a_3a_4-a_3^2)$	$(-a_3a_2-a_3a_4)$
$(-a_4a_1-a_4a_3)$	$(3a_4a_1-a_4^2)$	$(-a_4a_2-a_4a_3)$	$(3a_4a_3-a_4^2)$	$(3a_4a_2-a_4^2)$
$(4a_1^2+4a_1a_3+4a_2a_1+8a_2a_3)$	$(4a_1^2+4a_1a_4+4a_2a_1+8a_2a_4)$	$(4a_1a_2+8a_1a_3+4a_2^2+4a_2a_3)$	$(4a_1a_3+4a_1a_4+4a_2a_3+4a_2a_4)$	$(4a_1a_2+8a_1a_4+4a_2^2+4a_2a_4)$
$(8a_1^2+8a_1a_3+8a_3^2)$	$(4a_1^2+4a_1a_4+4a_1a_3+8a_3a_4)$	$(8a_1a_2+4a_1a_3+4a_3a_2+4a_3^2)$	$(8a_1a_4+4a_1a_3+4a_4a_3+4a_3^2)$	$(4a_1a_2+4a_1a_4+4a_3a_2+4a_3a_4)$
$(4a_1^2+4a_1a_3+4a_4a_1+8a_4a_3)$	$(8a_1^2+8a_1a_4+8a_4^2)$	$(4a_1a_2+4a_1a_3+4a_4a_2+4a_4a_3)$	$(4a_1a_4+8a_1a_3+4a_4^2+4a_3a_4)$	$(8a_1a_2+4a_1a_4+4a_4a_2+4a_4^2)$
$(8a_2a_1+4a_2a_3+4a_3a_1+4a_3^2)$	$(4a_2a_1+4a_2a_4+a_3a_1+4a_3a_4)$	$(8a_2^2+8a_2a_3+8a_3^2)$	$(4a_2a_3+8a_2a_4+4a_3^2+4a_3a_4)$	$(4a_2^2+4a_2a_4+4a_3a_2+8a_3a_4)$
$(4a_3a_1+4a_3^2+8a_4a_1+4a_3a_4)$	$(8a_3a_1+4a_3a_4+4a_4a_1+4a_4^2)$	$(4a_3a_2+4a_3^2+8a_4a_2+4a_4a_3)$	$(8a_3^2+8a_3a_4+8a_4^2)$	$(8a_3a_2+4a_3a_4+4a_4a_2+4a_4^2)$
$(4a_2a_1+4a_2a_3+4a_4a_1+4a_4a_3)$	$(8a_2a_1+4a_2a_4+4a_4a_1+4a_4^2)$	$(4a_2^2+4a_2a_3+4a_4a_2+8a_4a_3)$	$(8a_2a_3+4a_2a_4+4a_4a_3+4a_4^2)$	$(8a_2^2+8a_2a_4+8a_4^2)$

APPENDIX B

TWO DIMENSIONAL FINITE ELEMENT MODEL DERIVATION

GOVERNING EQUATION

The governing partial differential equation for the transient heat conduction model in two dimensions is

$$k_x \frac{\partial^2 t}{\partial x^2} + k_y \frac{\partial^2 t}{\partial y^2} = \rho c \frac{\partial t}{\partial \theta} \tag{1}$$

where (a) k_x, k_y are constant directional thermal conductivities

(b) θ is the time variable

(c) ρ is constant density

(d) c is the constant pressure specific heat

(e) an initial condition is given

(f) boundary values are specified

(g) $\partial t/\partial n = 0$ along the boundary of the area being studied (i.e., the subject area is thermally insulated)

Although Equation 1 assumes constant physical and thermal properties throughout the control area, variable properties can be handled by numerical methods proposed (see section entitled "Derivation of System Matrices", contained in this appendix).

DISCRETIZATION

Given an area A discretize A into a finite union of "m" triangular shaped elements $A^{(e)}$ where

$$A \approx \bigcup_{e=1}^{m} A^{(e)}$$

and the intersection of any two $A^{(e)}$ is an entire common edge of the triangles or a common vertex. For each triangle, specify nodal points at each vertex and at the midpoint of each edge. This will result

in 6 nodes for each element $A^{(e)}$. Due to the discretization of a continuous area, nearly all of the nodes will be shared by other elements. A special requirement of the discretization process is that two contiguous elements must share an entire common edge rather than a portion of the line segment. This must result in 3 common nodes for contiguous elements. Only when elements touch at a vertex will there be just one common node.

ELEMENT AND NODAL NUMBERING

Number the elements and nodes of the discretized area. Do not renumber shared nodes of contiguous elements. That is, if the system is discretized into two triangles contiguous along an edge, thre will be 2 elements and 9 nodes in the model, not 2 elements and 12 nodes.

DERIVATION OF SYSTEM MATRICES

It was shown by Desai and Abel (1972) that solving Equation 1 along with its specifications is equivalent to minimizing with respect to the variable "t" (temperature) the variational statement

$$I = \frac{1}{2} \iint \left(K_x \left(\frac{\partial t}{\partial x}\right)^2 + K_y \left(\frac{\partial t}{\partial y}\right)^2 + \rho c \frac{\partial t^2}{\partial \theta} \right) dA \qquad (2)$$

where we assume $\partial t/\partial n = 0$ along the area's boundary.

If we could minimize Equation 2 with respect to "t," we would arrive at an expression for temperature as a function of position and time. Instead of attempting to solve the above equation analytically, approximations will be used which when substituted into Equation 2 result in a system of linear equations with values of temperature as the unknowns.

The first step is to formulate a function for temperature within the area A. This will be done by specifying functions for temperature within each element $A^{(e)}$. These temperature functions will be continuous along the element edges such that Equation 2 will be defined.

Consider an element $A^{(e)}$ along with its six specified nodal points. Let

$$t = f(t_1, t_2, \ldots, t_6, x, y)$$

where "t" is the temperature at point "P," having coordinates (x,y), of the area A, and "t" is a function of the temperature of the element's nodal points, where "P" is contained in $A^{(e)}$, ($(t_1, t_2, \ldots, t_6)$ are the element's nodal temperatures in the order shown in Figure 7).

Using the "local cordinate" system defined in Chapter Five, we can write such an interpolating temperature function as

$$t = [(2L_1^2 - L_1)t_1 + (2L_2^2 - L_2)t_2 + (2L_3^2 - L_3)t_3 + 4L_1L_2t_4 + 4L_2L_3t_5 + 4L_3L_1t_6]$$

or in matrix notation $t = [\ N\ ]\underset{\sim}{t}^e$, or

$$t = [(2L_1^2 - L_1), (2L_2^2 - L_2), (2L_3^2 - L_3), 4L_1L_2, 4L_2L_3, 4L_3L_1]\ [t^{(e)}] \qquad (3)$$

where $[t^{(e)}]$ is the column vector of the nodal temperatures of $A^{(e)}$. This temperature function will equal the nodal temperature at each nodal point, and will interpolate between these nodal points to evaluate temperatures elsewhere within $A^{(e)}$.

Equation 3 is valid for each element $A^{(e)}$ within the area A. Also this equation represents a quadratic variation of "t" (temperature) throughout each $A^{(e)}$, providing increased accuracy over a linear interpolating temperature function.

The second step is to evaluate the integral expression shown in Equation 2. Assuming that

$$A = \bigcup_{e=1}^{m} A^{(e)}$$

and assuming Equation 2 can be rewritten as

$$I \approx \frac{1}{2} \iint_{\substack{m \\ \cup \\ e=1}} A^{(e)} \left(k_x \left(\frac{\partial t}{\partial x}\right)^2 + k_y \left(\frac{\partial t}{\partial t}\right)^2 + \rho c \frac{\partial t^2}{\partial \theta}\right) dA \tag{4}$$

then Equations 2 and 4 can be combined and expanded into

$$I \approx \sum_{e=1}^{m} \frac{1}{2} \iint_{A^{(e)}} \left(k_x \left(\frac{\partial t}{\partial x}\right)^2 + k_y \left(\frac{\partial t}{\partial y}\right)^2 + \rho c \frac{\partial t^2}{\partial \theta}\right) dA \tag{5}$$

Thus we can consider each element $A^{(e)}$ individually in evaluating 5, such that

$$I \approx \sum_{e=1}^{m} I^{(e)} \tag{6}$$

where

$$I^{(e)} = \frac{1}{2} \iint \left(k_x \left(\frac{\partial t}{\partial x}\right)^2 + k_y \left(\frac{\partial t}{\partial y}\right)^2 + \rho c \frac{\partial t^2}{\partial \theta}\right) dA$$

The third step is to note that since we have an interpolating temperature function that is valid for each element $A^{(e)}$, we can approximate the partial derivatives in Equation 6 by the differential formulae established in Chapter 5. Furthermore, these expressions will be valid for each element.

To simplify calculations, rewrite the expression for $I^{(e)}$ in Equation 6 as

$$I^{(e)} = \frac{1}{2} \iint_{A^{(e)}} k_x \left(\frac{\partial t}{\partial x}\right)^2 dA + \frac{1}{2} \iint_{A^{(e)}} k_y \left(\frac{\partial t}{\partial y}\right)^2 dA + \frac{1}{2} \iint_{A^{(e)}} \rho c \frac{\partial t^2}{\partial \theta} \, dA \tag{7}$$

or simply

$$I^{(e)} = I_x^{(e)} + I_y + I_{\rho c} \tag{8}$$

where the subscripts refer to the identifying governing parameter for each expression in Equation 7. The reason for this is $I_x^{(e)}$ and $I_y^{(e)}$ are similar, except for some multiplying constants of position and thermal parameters. Hence, we only need to find an useable expression for $I_x^{(e)}$, and then expand the results to include $I_y^{(e)}$. $I_{\rho c}^{(e)}$ must be solved separately due to the time derivative.

Our problem is to minimize Equation 2 with respect to "t" (temperature). This is equivalent to minimizing Equation 4 with respect to temperature. But our interpolating temperature function is a function of nodal temperatures. Thus we must minimize equation 5 with respect to each nodal tempoerature of the entire area A. By Equation 6, we see that we must minimize each $I^{(e)}$ with respect to each nodal temperature of A. Since only six nodes are contained in each $A^{(e)}$, most nodal temperature minimization efforts (for a several-element model) equate to zero. That is, minimizing Equation 4 with respect to each of the area's A nodal temperatures is equivalent to the sum of the minimizations of $I^{(e)}$ with respect to each nodal temperature of $A^{(e)}$.

Hence, for each $A^{(e)}$ we want to evaluate

$$\frac{\partial I^{(e)}}{\partial t_1} = 0, \quad \frac{\partial I^{(e)}}{\partial t_2} = 0, \ldots, \quad \frac{\partial I^{(e)}}{\partial t_6} = 0$$

where t_i is the nodal temperature as expressed in Equation 3.

Since

$$I^{(e)} = I_x^{(e)} + I_y^{(e)} + I_{\rho c}^{(e)} \tag{9}$$

we can minimize $I_x^{(e)}$ with respect to $(t_1, t_2, \ldots, t_6)$ of $A^{(e)}$, and then extend these results to the minimization of $I_y^{(e)}$. However, $I_{\rho c}^{(e)}$ must be minimized with respect to $(t_1, t_2, \ldots, t_6)$ of $A^{(e)}$ separately.

Step four is performing the above described operations. We will use the notation a_i and b_i for the cofactors of the element's coordinate matrices (see below) and Equation 3 for the interpolating temperaturefunction.

Definition:

Let $a_1 = x_3 - x_2$ $\quad b_1 = y_2 - y_3$

$a_2 = x_1 - x_3$ $\quad b_2 = y_3 - y_1$

$a_3 = x_2 - x_1$ $\quad b_3 = y_1 - y_2$

where (x_i, y_i) are the coordinates of nodal point location "i" of the two dimensional triangular element.

Consider for any $A^{(e)}$, $I_x^{(e)} = \frac{1}{2} k_x \iint_{A^{(e)}} \left(\frac{\partial t}{\partial x}\right)^2 dA$

then

$$\frac{\partial I_x^{(e)}}{\partial t_1} = k_x \iint_{A^{(e)}} \left(\frac{\partial t}{\partial x}\right) \frac{\partial \left(\frac{\partial t}{\partial x}\right)}{\partial t_1} dA$$

$$= k_x \iint_{A^{(e)}} \left(\frac{\partial t}{\partial x}\right) \frac{b_1}{2A^{(e)}} (4L_1 - 1)\, dA$$

$$= \frac{b_1 k_x}{(2A^{(e)})^2} \iint_{A^{(e)}} \{ b_1 [(16L_1^2 - 8L_1 + 1)t_1 + (4(4L_1L_2 - L_2))t_4 + 4(4L_1L_3 - L_3)t_6]$$

$$+ b_2 [(16L_2L_1 - 4L_2 - 4L_1 + 1)t_2 + 4(4L_1^2 - L_1)t_4 + 4(4L_1L_3 - L_3)t_5]$$

$$+ b_3 [(16L_3L_1 - 4L_3 - 4L_1 + 1)t_3 + 4(4L_1L_2 - L_2)t_5 + 4(4L_1^2 - L_1)t_6] \} dA$$

where integration is solved as follows:

$$\iint_{A^{(e)}} L_1^p L_2^q L_3^r \, dA = (p!q!r!)/(p+q+r+2)!$$

Hence,

i) $\iint_{A^{(e)}} dA = A^{(e)}$

ii) $\iint_{A^{(e)}} L_1^2 \, dA = A^{(e)}\left(\frac{1}{6}\right)$

iii) $\iint_{A^{(e)}} L_1 \, dA = A^{(e)}\left(\frac{1}{3}\right)$

iv) $\iint_{A^{(e)}} L_1 L_2 \, dA = A^{(e)}\left(\frac{1}{12}\right)$

$$\therefore \frac{\partial I_x^{(e)}}{\partial t_1} = \frac{b_1 k_x}{4A^{(e)}}\left\{b_1 t_1 + b_2\left[\tfrac{-1}{3}t_2 + \tfrac{4}{3}t_4\right] + b_3\left[\tfrac{-1}{3}t_3 + \tfrac{4}{3}t_6\right]\right\}$$

The other partial derivatives follow:

$$\frac{\partial I_x^{(e)}}{\partial t_2} = \frac{b_2 k_x}{4A^{(e)}}\left\{b_1\left(\tfrac{-1}{3}t_1 + \tfrac{4}{3}t_4\right) + b_2 t_2 + b_3\left(\tfrac{-1}{3}t_3 + \tfrac{4}{3}t_6\right)\right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_3} = \frac{b_3 k_x}{4A^{(e)}}\left\{b_1\left(\tfrac{-1}{3}t_1 + \tfrac{4}{3}t_6\right) + b_2\left(\tfrac{-1}{3}t_2 + \tfrac{4}{3}t_5\right) + b_3 t_3\right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_4} = \frac{k_x}{A^{(e)}}\left\{\frac{b_2 b_1}{3}t_1 + \frac{b_1 b_2}{3}t_2 + \frac{2}{3}(b_2^2 + b_1 b_2 + b_1^2)t_4 + \frac{1}{3}(b_2^2 + b_2 b_3 + b_1 b_2 + 2b_1 b_3)t_5 + \frac{1}{3}(b_2 b_1 + 2b_2 b_3 + b_1^2 + b_1 b_3)\right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_5} = \frac{k_x}{A^{(e)}}\left\{\frac{b_2 b_3}{3}t_2 + \frac{b_2 b_3}{3}t_3 + \frac{1}{3}(2b_3 b_1 + b_2 b_3 + b_2 b_1 + b_2^2)t_4 + \frac{1}{3}(b_3 b_2 + 2b_3^2 + 2b_2^2 + b_2 b_3)t_5 + \frac{1}{3}(b_3 b_1 + b_3^2 + 2b_2 b_1 + b_2 b_3)t_6\right\}$$

$$\frac{\partial I_x^{(e)}}{\partial t_6} = \frac{k_x}{A^{(e)}}\left\{\frac{b_3 b_1}{3}t_1 + \frac{b_1 b_3}{3}t_3 + \frac{1}{3}(b_1^2 + b_1 b_2 + b_3 b_1 + b_3 b_2)t_4 + \frac{1}{3}(2b_1 b_2 + b_1 b_3 + b_3 b_2 + b_3^2)t_5 + \frac{1}{3}(2b_1^2 + b_1 b_3 + b_3 b_1 + 2b_3^2)t_6\right\}$$

The preceeding processes of partial differentiation must be repeated for $I_y^{(e)}$. However, parallel development will result in the same expressions as derived for $I_x^{(e)}$, except that for $I_y^{(e)}$ we must substitute a_i for b_i, and k_y for k_x.

Let $\underset{\sim}{t}^{(e)}$ = the column vector $[t_1, t_2, \ldots, t_6]^T$. Then minimizing $I_x^{(e)} + I_y^{(e)}$ with respect to $(t_1, t_2, \ldots, t_6)$ of $A^{(e)}$ is equivalent to

$$\frac{\partial (I_x^{(e)} + I_y^{(e)})}{\partial \underset{\sim}{t}^{(e)}} = 0 \tag{10}$$

By the above derivations, we see that Equation 9 establishes the matrix expression

$$[X \quad K^{(e)}] \underset{\sim}{t}^{(e)} = 0 \tag{11}$$

where $[X \quad K^{(e)}]$ is the "element conduction matrix" for $A^{(e)}$, where $\underset{\sim}{t}^{(e)}$ is the column vector of nodal temperatures of $A^{(e)}$.

Note that we can expand Equation 10 as

$$[X \quad K^{(e)}]\underset{\sim}{t}^{(e)} = \{[\ K_x^{(e)}] + [\ K_y^{(e)}]\}\underset{\sim}{t}^{(e)} = 0$$

where each matrix in the above sum corresponds to the minimization of its respective entry in Equation 9. The matrix $[\ K_x^{(e)}]$ is written in matrix form in the above derivation. The matrix $[\ K_y^{(e)}]$ can be found by substituting the constants a_i for the b_i, and k_y for k_x.

Equation 11 only represents part of the expression for minimizing Equation 7. We must still minimize $I_{\rho c}^{(e)}$ with respect to $(t_1, t_2, \ldots, t_6)$ of $A^{(e)}$, that is we must solve

$$\frac{\partial I_{\rho c}^{(e)}}{\partial \underset{\sim}{t}^{(e)}} = 0$$

Consider for any $A^{(e)}$, $I_{\rho c}^{(e)} = \frac{1}{2}\iint_{A^{(e)}} (\rho c)^{(e)} \frac{\partial t^2}{\partial \theta}\, dA$, θ = time;

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_1} = (\rho c)^{(e)} \frac{d}{d\theta} \iint_{A^{(e)}} [N]\, \underset{\sim}{t}^{(e)} (2L_1^2 - L_1)\, dA$$

$$= 2(\rho c A)^{(e)} \frac{d}{d\theta}\left[\frac{1}{60}t_1 + \frac{-1}{360}t_2 + \frac{-1}{360}t_3 + \frac{-1}{90}t_5\right]$$

The other partial derivatives follow:

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_2} = 2(\rho c A)^{(e)} \frac{d}{d\theta}\left[\frac{-1}{360}t_1 + \frac{1}{60}t_2 + \frac{-1}{360}t_3 + \frac{-1}{90}t_6\right]$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_3} = 2(\rho c A)^{(e)} \frac{d}{d\theta}\left[\frac{-1}{360}t_1 + \frac{-1}{360}t_2 + \frac{1}{60}t_3 + \frac{-1}{90}t_6\right]$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_4} = 2(\rho c A)^{(e)} \frac{d}{d\theta}\left[\frac{-1}{90}t_3 + \frac{4}{45}t_4 + \frac{2}{45}t_5 + \frac{2}{45}t_6\right]$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_5} = 2(\rho c A)^{(e)} \frac{d}{d\theta}\left[\frac{-1}{90}t_1 + \frac{2}{45}t_4 + \frac{4}{45}t_5 + \frac{2}{45}t_6\right]$$

$$\frac{\partial I_{\rho c}^{(e)}}{\partial t_6} = 2(\rho c A)^{(e)} \frac{d}{d\theta}\left[\frac{-1}{90}t_2 + \frac{2}{45}t_4 + \frac{2}{45}t_5 + \frac{4}{45}t_6\right]$$

or in matrix notation $\frac{\partial I_{\rho c}^{(e)}}{\partial \underset{\sim}{t}^{(e)}} =$

$$2(\rho c A)^{(e)} \begin{bmatrix} \frac{1}{60} & \frac{1}{360} & \frac{-1}{360} & 0 & \frac{-1}{90} & 0 \\ \frac{-1}{360} & \frac{1}{60} & \frac{-1}{360} & 0 & 0 & \frac{-1}{90} \\ \frac{-1}{360} & \frac{-1}{360} & \frac{1}{60} & \frac{-1}{90} & 0 & 0 \\ 0 & 0 & \frac{-1}{90} & \frac{4}{45} & \frac{2}{45} & \frac{2}{45} \\ \frac{-1}{90} & 0 & 0 & \frac{2}{45} & \frac{4}{45} & \frac{2}{45} \\ 0 & \frac{-1}{90} & 0 & \frac{2}{45} & \frac{2}{45} & \frac{4}{45} \end{bmatrix} \dot{\underset{\sim}{t}}^{(e)}$$

$$\frac{\partial I_x^{(e)}}{\partial \underset{\sim}{t}^{(e)}} = \frac{k_x}{12 A^{(e)}} \begin{bmatrix}
3b_1^2 & -b_1b_2 & -b_1b_3 & 4b_1b_2 & 0 & 4b_1b_3 \\
-b_1b_2 & 3b_2^2 & -b_2b_3 & 4b_1b_2 & 4b_2b_3 & 0 \\
-b_1b_3 & -b_2b_3 & 3b_3^2 & 0 & 4b_2b_3 & 4b_1b_3 \\
4b_1b_2 & 4b_1b_2 & 0 & 8(b_2^2+b_1b_2+b_1^2) & 4(b_2^2+b_2b_3 \\ & & & & +b_1b_2+2b_1b_3) & 4(b_1b_2+2b_2b_3 \\ & & & & & +b_1^2+b_1b_3) \\
0 & 4b_2b_3 & 4b_2b_3 & 4(2b_1b_3+b_2b_3 & 8(b_3^2+b_2b_3 & 4(b_1b_3+b_3^2 \\ & & & +b_1b_2+b_2^2) & +b_2^2) & +2b_1b_2+b_2b_3) \\
4b_1b_3 & 0 & 4b_1b_3 & 4(b_1^2+b_1b_2 & 4(b_3^2+2b_1b_2 & 8(b_1^2+b_1b_3+b_3^2) \\ & & & +b_1b_3+b_2b_3) & +b_1b_3+b_2b_3) &
\end{bmatrix}$$

Minimizing $I_{\rho c}^{(e)}$ with respect to $(t_1, t_2, \ldots, t_6)$ of $A^{(e)}$ results in the "element capacitance matrix" expression

$$[XC^{(e)}]\,\dot{\underset{\sim}{t}}^{(e)} = 0 \tag{12}$$

What we wanted to do was minimize Equation 7 with respect to $(t_1, t_2, \ldots, t_6)$ of $A^{(e)}$. Hence, combining Equations 7, 10 and 11 we get

$$[X\ \ K^{(e)}]\underset{\sim}{t}^{(e)} + [XC^{(e)}]\dot{\underset{\sim}{t}}^{(e)} = 0 \tag{13}$$

By equations 5 and 6, we must perform the above operations for each of the 'm' elements, and sum the results in matrix form. This combination of element matrices will form the "global matrix" system

$$[G\ \ K]\underset{\sim}{t} + [GC]\dot{\underset{\sim}{t}} = 0 \tag{14}$$

where

$[G\ \ K]$ = the global conduction matrix

$[GC]$ = the global capacitance matrix

$\underset{\sim}{t}$ = the column vector of all nodal temperatures

$\dot{\underset{\sim}{t}}$ = the column vector of time derivatives of all nodal temperatures

Using Equation 13, we can apply a time advancement modification such as the "Crank-Nicolson" method, and formulate a matrix system that can be programmed into the computer. From the above it can be seen that variable physical and thermal parameters can be handled by specifying constant parameters per each element $A^{(e)}$.

APPENDIX C

ELEMENT PHASE CHANGE, GLOBAL MATRIX UPDATING

When is the element considered changed of phase? In the proposed model, nodes are allowed to change chase independent of the associated element's phases. Additionally, we can have frozen nodes in a thawed element, or thawed nodes in a frozen element.

The node changes phase when the requisite amount of latent heat is appropriately supplied or evolved. However, such a scheme for an element phase change process is inadequate in the critical region of the freezing front where all nodes could be considered "frozen," and yet actually be in the transitory stage of phase change where less than the total amount of latent heat necessary to be completely frozen has been evolved, which could keep the element thawed. The model uses the simple scheme of keeping track of the number of nodes thawed for each element.

Each element is flagged to indicate whether the element is thawed or frozen. Additionally, a vector NTHAWD is used to store the number of nodes in the thawed state for each element.

On input, the phase of each element is flagged on a stored thermal parameter. For tetrahedron or triangular elements the latent heat per unit mass is flagged; for Brick elements, the NTHAWD(i) vector is flagged

(a) a positive parameter implies the element is thawed (contains a positive amount of latent heat)

(b) a negative parameter implies the element is frozen.

The Crank-Nicolson method modified the global matrix system. To update this modified system, the phase-change element matrices must be likewise modified. Using previous notation, the global conduction matrix $\underset{\sim}{G}K$ and the

global capacitance matrix $\underset{\sim}{GC}$ are modified as

$$\underset{\sim}{GK}^* = \underset{\sim}{GK} + \frac{2}{\Delta\theta}\,\underset{\sim}{GC}$$

$$\underset{\sim}{GC}^* = \frac{2}{\Delta\theta}\,\underset{\sim}{GC} - \underset{\sim}{GK}$$

where after computation, $\underset{\sim}{GK}^*$ and $\underset{\sim}{GC}^*$ replaces the matrices $\underset{\sim}{GK}$ and $\underset{\sim}{GC}$ in computer memory.

By definition, the global matrices $\underset{\sim}{GK}$ and $\underset{\sim}{GC}$ are sums of the appropriate element matrices $\underset{\sim}{XK}^{(e)}$ and $\underset{\sim}{XC}^{(e)}$. The following example demonstrates how to update the global matrix system due to a phase change of an element, without rederiving the entire system matrices. Let

$$\underset{\sim}{GK} = \underset{\sim}{A} + \underset{\sim}{B} + \underset{\sim}{C}$$

$$\underset{\sim}{GC} = \underset{\sim}{D} + \underset{\sim}{E} + \underset{\sim}{F}$$

Suppose that during the simulation, an element changes phase, transforming element matrices $\underset{\sim}{C}$ and $\underset{\sim}{F}$ into matrices $\underset{\sim}{X}$ and $\underset{\sim}{Y}$ respectively. The global matrix modifications from implementation of the Crank-Nicolson method would have previously changed $\underset{\sim}{GK}$ and $\underset{\sim}{GC}$ such that

$$\underset{\sim}{GK}^* = (\underset{\sim}{A} + \underset{\sim}{B} + \underset{\sim}{C}) + \frac{2}{\Delta\theta}\,(\underset{\sim}{D} + \underset{\sim}{E} + \underset{\sim}{F})$$

$$GC^* = (-\underset{\sim}{A} - \underset{\sim}{B} - \underset{\sim}{C}) + \frac{2}{\Delta\theta}\,(\underset{\sim}{D} + \underset{\sim}{E} + \underset{\sim}{F})$$

where $\underset{\sim}{GK}^*$ and $\underset{\sim}{GC}^*$replaced $\underset{\sim}{GK}$ and $\underset{\sim}{GC}$ in computer memory. Duplicate the above transformations on the element matrix changes

$$(\underset{\sim}{X} - \underset{\sim}{C})^* = (\underset{\sim}{X} - \underset{\sim}{C}) + \frac{2}{\Delta\theta}\,(\underset{\sim}{Y} - \underset{\sim}{F})$$

$$(\underset{\sim}{Y} - \underset{\sim}{F})^* = (-\underset{\sim}{X} + \underset{\sim}{C}) + \frac{2}{\Delta\theta}\,(\underset{\sim}{Y} - \underset{\sim}{F})$$

Now add $(X - C)^*$ and $(\underset{\sim}{Y} - \underset{\sim}{F})^*$ to $\underset{\sim}{GK}^*$ and $\underset{\sim}{GC}^*$respectively to the above

$$\underset{\sim}{GK}^* + (\underset{\sim}{X} - \underset{\sim}{C})^* = (\underset{\sim}{A} + \underset{\sim}{B} + \underset{\sim}{X}) + \frac{2}{\Delta\theta}(\underset{\sim}{D} + \underset{\sim}{E} + \underset{\sim}{Y})$$

$$\underset{\sim}{GC}^* + (\underset{\sim}{Y} - \underset{\sim}{F})^* = (-\underset{\sim}{A} - \underset{\sim}{B} - \underset{\sim}{X}) + \frac{2}{\Delta\theta}(\underset{\sim}{D} + \underset{\sim}{E} + \underset{\sim}{Y})$$

which shows that the global matrix system has been properly modified to accommodate the updating of an element matrix due to phase change.

The following example of a two element system undergoing "phase change" will demonstrate the above.

Let the modified element matrices for the thawed phase be

$$\underset{\sim}{XK}^* = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \qquad \underset{\sim}{XC}^* = \begin{bmatrix} -4 & 3 \\ 3 & -4 \end{bmatrix}$$

and for the frozen phase

$$\underset{\sim}{XK}^* = \begin{bmatrix} 3 & 4 \\ 4 & 3 \end{bmatrix} \qquad \underset{\sim}{XC}^* = \begin{bmatrix} 2 & 8 \\ 8 & 2 \end{bmatrix}$$

Assume both elements are initially thawed, then the system is

$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 2 & 2 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i+1} = \begin{bmatrix} -4 & 3 & 0 \\ 3 & -8 & 3 \\ 0 & 3 & -4 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i}$$

where "i" refers to a time step interval, and t_1, t_2, t_3 are nodal temperatures. Assume that a boundary condition is $t_1 = \eta$. Then

$$\underset{GK^*}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 0 & 2 & 1 \end{bmatrix}} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i+1} = \underset{GC^*}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & -8 & 3 \\ 0 & 3 & -4 \end{bmatrix}} \begin{bmatrix} \eta \\ t_2 \\ t_3 \end{bmatrix} + \underset{TT^*}{\begin{bmatrix} 0 \\ -\eta(2-3) \\ -\eta(0-0) \end{bmatrix}} \qquad (1)$$

Now, consider what the system would be had element number one been initially frozen. The system would define the following relation:

$$\begin{bmatrix} 3 & 4 & 0 \\ 4 & 4 & 2 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i+1} = \begin{bmatrix} 2 & 8 & 0 \\ 8 & -2 & 3 \\ 0 & 3 & -4 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i}$$

where $t_1 = \eta$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 2 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 3 \\ 0 & 3 & -4 \end{bmatrix} \begin{bmatrix} \eta \\ t_2 \\ t_3 \end{bmatrix}^{i} + \begin{bmatrix} 0 \\ -\eta(4-8) \\ -\eta(0-0) \end{bmatrix} \qquad (2)$$

$\underset{\sim}{GK}^*$ $\underset{\sim}{GC}^*$ $\underset{\sim}{TT}^*$

If the first system shown in Equation 1 was in progress, the element one changes phase, then after updating, the system should be represented by Equation 2. First, the change in element matrices in calculated as follows:

$$\begin{bmatrix} 3 & 4 & 0 \\ 4 & 3 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

which is the change for matrix $\underset{\sim}{GK}^*$ in Equation 1, and

$$\begin{bmatrix} 2 & 8 & 0 \\ 8 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} -4 & 3 & 0 \\ 3 & -4 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 6 & 5 & 0 \\ 5 & 6 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

which is the change for matrix $\underset{\sim}{GC}^*$ in Equation 1.

Therefore, Equation 1 becomes

$$\begin{bmatrix} 3 & 2 & 0 \\ 2 & 4 & 2 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i+1} = \begin{bmatrix} 7 & 5 & 0 \\ 5 & -2 & 3 \\ 0 & 3 & -4 \end{bmatrix} \begin{bmatrix} \eta \\ t_2 \\ t_3 \end{bmatrix}^{i} + \begin{bmatrix} 0 \\ -\eta(2-3) \\ -\eta(0-0) \end{bmatrix} \qquad (3)$$

Reapplying the boundary condition $t_1 = \eta$, Equation 3 becomes

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 2 \\ 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i+1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -2 & 3 \\ 0 & 3 & -4 \end{bmatrix} \begin{bmatrix} \eta \\ t_2 \\ t_3 \end{bmatrix}^{i} + \begin{bmatrix} 0 \\ -\eta(2-3)-\eta(2-5) \\ -\eta(0-0) - \eta(0-0) \end{bmatrix}$$

or

$$\underset{\sim}{GK^*} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}^{i+1} = \underset{\sim}{GC^*} \begin{bmatrix} \eta \\ t_2 \\ t_3 \end{bmatrix}^{i} + \underset{\sim}{TT} \begin{bmatrix} 0 \\ -\eta(4-8) \\ -\eta(0-0) \end{bmatrix}$$

which is exactly the relation stated in Equation 2.

In conclusion, if an element changes phase, the following procedure is used:

(1) calculate the change in thermal parameters, i.e. parameter (new) - parameter (old).

(2) using the change in parameters as the thermal parameter, rederive the element matrices, modify the results per the Crank-Nicolson modifications, and add to the global matrix system.

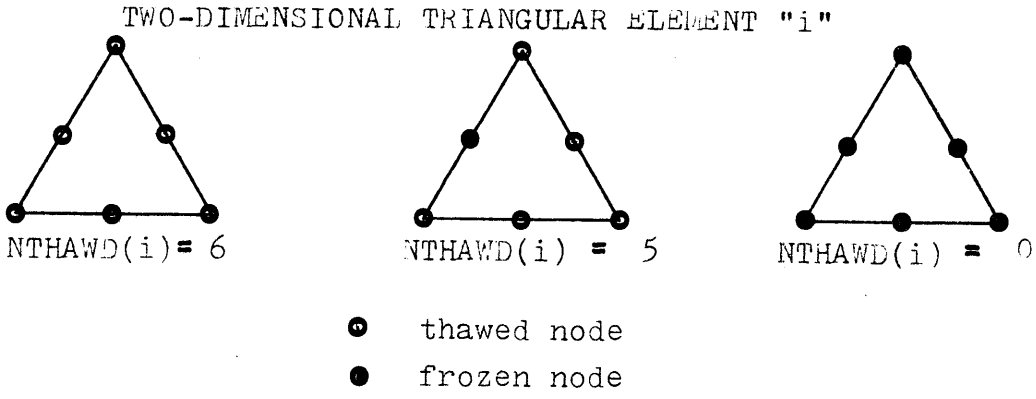
(3) reapply the boundary conditions.

(4) update element phase flagging whether frozen or thawed.

The above procedure, however, does not consider the changes imposed on $\underset{\sim}{GK^*}$ and $\underset{\sim}{GC^*}$ by the insertion of boundary conditions, which occurs prior to the time advancement routine.

If the element which changes phase is involved with columns α and β, then we must also modify $\underset{\sim}{TT}$ for the new thermal parameters.

Intuitively, columns α and β represent the columns of summed matrices. Hence, by the summing of matrices $(\underset{\sim}{X}-\underset{\sim}{C})^*$ and $(\underset{\sim}{Y}-\underset{\sim}{F})^*$ to $\underset{\sim}{G}\underset{\sim}{K}^*$ and $\underset{\sim}{G}\underset{\sim}{C}^*$, (after the boundary conditions were already inserted in $\underset{\sim}{G}\underset{\sim}{K}^*$ and $\underset{\sim}{G}\underset{\sim}{C}^*$), we see that only a column of $(\underset{\sim}{X}-\underset{\sim}{C})^*$ or $(\underset{\sim}{Y}-\underset{\sim}{F})^*$ values would show in α and β . If we were to reapply the boundary condition insertion scheme, the vector $\underset{\sim}{T}\underset{\sim}{T}$ would be properly adjusted.



NTHAWD (i) is adjusted as follows:

a) if node A freezes, add -1 to NTHAWD(i)

b) if node A thaws, add +1 to NTHAWD(i)

where "i" represents all elements associated to node A .

Thus, if a node A changes phase, each element "i" containing node A is determined, NTHAWD (i) is updated, and

a) if NTHAWD(i)= 0 , and the element "i" is already frozen: no phase change.

b) if NTHAWD(i)= 0 , and the element "i" is already thawed: element "i" freezes.

c) if NTHAWD(i)= 6 , and the element "i" is already frozen: element "i" thaws.

d) if NTHAWD(i)= 6 , and the element "i" is already thawed: no phase change.

Hence, if NTHAWD(i) is between 0 and 6 , there is no need to even test for phase change. (For three-dimensional problems, the critical numbers are NTHAWD(i) = 0 and NTHAWD(i) = 10).

FIGURE C-1

APPENDIX D

USER'S MANUAL

TABLE OF CONTENTS

FIRST CONTROL CARD (I2)

cc. 2	KODE: DIMENSION OF PROBLEM (2)

SECOND CONTROL CARD (I1)

cc. 1	KODE 1: "2" IS FOR TRANSIENT HEAT CONDUCTION MODEL (ANISOTROPIC)

FIRST DATA CARD SET (6I4, 2F10.5)

cc. 1-4	NBAND: BANDWIDTH OF GLOBAL MATRICES
cc. 5-8	NNODES: NUMBER OF NODES IN PROBLEM
cc. 9-12	NELE: TOTAL NUMBER OF ELEMENTS
cc. 13-16	NBRICK: NUMBER OF BRICK ELEMENTS
cc. 17-20	NTETRA: NUMBER OF TETRAHEDRON ELEMENTS
cc. 21-24	NUMBC: NUMBER OF NODES WITH SPECIFIED TEMPERATURES
cc. 25-34	THETA: TIME STEP, EXPRESSED IN HOURS
cc. 35-44	DAYS: DURATION OF TEST, EXPRESSED IN HOURS

SECOND DATA CARD SET (6F11.5)

cc. 1-66	BC(I): TEMPERATURES OF BOUNDARY CONDITIONS, EXPRESSED IN ^{o}F., READ IN NUMERICAL ORDER-OF-MAGNITUDE OF CORRESPONDING GLOBAL NODE NUMBERS.

THIRD DATA CARD SET(11I6)

cc. 1-66	NBC(I): GLOBAL NODE NUMBERS OF BOUNDARY CONDITION, READ IN NUMBERICAL ORDER-OF-MAGNITUDE.

FOURTH DATA CARD SET (12I6,/,12I6,/,3I6) IBRICK (I,J)

(OMIT IF THERE ARE NO BRICK ELEMENTS, I.E. NBRICK= 0)

INPUT BRICK NODAL SEQUENCE, IN ELEMENT ORDER OF MAGNITUDE.

NOTE: BRICK NODAL INPUT MUST BE IN "STANDARD NODAL SEQUENCE."

DATA INPUT IS PUNCHED ON THREE CARDS FOR EACH BRICK ELEMENT INTO THE MATRIX IBRICK (I,J) AS FOLLOWS:

CARD ONE: (12I6)

cc. 1-72 FIRST TWELVE NODES IN SEQUENCE.

CARD TWO: (12I6)

cc. 1-72 NEXT TWELVE NODES IN SEQUENCE.

CARD THREE: (3I6)

cc. 1-12 LAST TWO NODES IN SEQUENCE

cc. 13-18 ELEMENT NUMBER

(NOTE: REPEAT THIS PROCEDURE FOR EACH BRICK ELEMENT.)

FIFTH DATA CARD SET(6F10.4,/4F10.4) BRICK (I,J)

(OMIT IF THERE ARE NO BRICK ELEMENTS, I.E., IF NBRICK= 0)

INPUT BRICK PARAMETER DATA FOR EACH BRICK INTO MATRIX BRICK (I,J) AS FOLLOWS:

CARD ONE: (6F10.4)

cc. 1-10 X(1) : GLOBAL X-COORDINATES OF BRICK ELEMENT NODE CORRESPONDING TO STANDARD BRICK NODE NUMBER ONE (FT)

cc. 11-20 Y(1): GLOBAL Y-COORDINATE OF SAID NODE. (FT)

cc. 21-30 Z(1): GLOBAL Z-COORDINATE OF SAID NODE (FT)

cc. 31-40 DELX: X-DIMENSION OF BRICK (FT)

cc. 41-50 DELY: Y-DIMENSION OF BRICK (FT)

cc. 51-60 DELZ: Z-DIMENSION OF BRICK (FT)

CARD TWO: (4F10.4)

cc. 1-10 XKX: THERMAL CONDUCTIVITY IN X DIRECTION (BTU/HR. FT. - ^{o}F)

cc. 11-20 XKY: THERMAL CONDUCTIVITY IN DIRECTION OF Y (BTU/HR -ft.- ^{o}F)

cc. 21-30 XKZ: THERMAL CONDUCTIVITY IN Z DIRECTION (BTU/HR -FT.- ^{o}F)

cc. 31-40 THER: PRODUCT OF ELEMENT'S DENSITY (LBM/FT^3) AND SPECIFIC HEAT (BTU/LBM - ^{o}F)

(NOTE: REPEAT THIS PROCEDURE FOR EACH BRICK ELEMENT.)

SIXTH DATA CARD SET (11I6) ITETRA (I,J)

(OMIT IF THERE ARE NO TETRAHEDRON ELEMENTS: I.E. IF NTETRA = 0)

INPUT TETRAHEDRON NODAL SEQUENCE AND TETRAHEDRON ELEMENT NUMBER. NOTE: NODAL INPUT MUST BE IN "STANDARD NODAL SEQUENCE"

CARD ONE: (11I6)

cc. 1-60 TEN NODES IN ABOVE SEQUENCE

cc. 61-66 ELEMENT NUMBER.

(NOTE: REPEAT THIS PROCEDURE FOR EACH TETRAHEDRON ELEMENT.)

SEVENTH DATA CARD SET (6F10.4,/6F10.4,/.4F10.4, TETRA (I,J))

(OMIT IF THERE ARE NO TETRAHEDRON ELEMENTS: I.E. IF NTETRA = 0)

INPUT TETRAHEDRON PARAMETER DATA FOR EACH TETRAHEDRON ELEMENT INTO MATRIX (I,J) AS FOLLOWS:

CARD ONE: (6F10.4)

cc. 1-60 X(1), Y(1), Z(1), X(2), Y(2), Z(2)

CARD TWO:(6F10.4)

cc. 1-60 X(3), Y(3), Z(3), X(4), Y(4), Z(4),

CARD THREE(4F10.4)

cc. 1-10 XKX: THERMAL CONDUCTIVITY IN X DIRECTION (BTU/HR-FT$^{\circ}$F)

cc. 11-20 XKY: " " Y "

cc. 21-30 XKZ: " " Z "

cc. 31-40 THER: PRODUCT OF DENSITY (LBM/FT^3) AND SPECIFIC HEAT (BTU/LBM-$^{\circ}$F)

(NOTE: REPEAT FOR EACH TETRAHEDRON ELEMENT)

EIGHTH DATA CARD SET (6F11.5) BEGIN (I)

cc. 1-66 FIRST SIX INITIAL TEMPERATURES.($^{\circ}$F). INPUT MUST BE IN NODAL ORDER OF MAGNITUDE. INPUT ALL INITIAL TEMPERATURES, (INCLUDING BOUNDARY CONDITIONS), FOR I = 1 TO NNODES.

FIRST CONTROL CARD (I2)

cc. 2	KODE: DIMENSION OF PROBLEM (2)

SECOND CONTROL CARD (I1)

cc. 1	KODE1: "2" IS FOR TRANSIENT HEAT CONDUCTION MODEL (ANISOTROPIC)

FIRST DATA CARD SET (6F11.5)

cc. 1-4	NBAND: BANDWIDTH OF GLOBAL MATRICES
cc. 5-8	NNODES: NUMBER OF NODES IN PROBLEM
cc. 9-12	NELE: TOTAL NUMBER OF ELEMENTS
cc. 13-16	NBRICK: NUMBER OF BRICK ELEMENTS
cc. 17-20	NTETRA: NUMBER OF TETRAHEDRON ELEMENTS
cc. 21-24	NUMBC: NUMBER OF NODES WITH SPECIFIED TEMPERATURES
cc. 25-34	THETA: TIME STEP, EXPRESSED IN HOURS
cc. 35-44	DAYS: DURATION OF TEST, EXPRESSED IN HOURS

SECOND DATA CARD SET (6F11.5)

cc. 1-66	BC(I): TEMPERATURES OF BOUNDARY CONDITIONS, EXPRESSED IN ^{o}F., READ IN NUMERICAL ORDER - OF - MAGNITUDE OF CORRESPONDING GLOBAL NODE NUMBERS.

THIRD DATA CARD SET (11I6)

cc. 1-66	NBC(I): GLOBAL NODE NUMBERS OF BOUNDARY CONDITION, READ IN NUMERICAL ORDER-OF-MAGNITUDE.

FOURTH DATA CARD SET (10I6) IBRICK (I,J)

(OMIT IF THERE ARE NO BRICK ELEMENTS, I.E. NBRICK = 0)

INPUT BRICK NODAL SEQUENCE, IN ELEMENT ORDER OF MAGNITUDE.

NOTE: BRICK NODAL INPUT MUST BE IN "STANDARD NODAL SEQUENCE".

cc. 1-54 NINE NODES OF SEQUENCE

cc. 55-60 ELEMENT NUMBER

FIFTH DATA CARD SET (7F10.4) BRICK (I,J)

(OMIT IF THERE ARE NO BRICK ELEMENTS, I.E. NBRICK = 0)

INPUT BRICK PARAMETER DATA FOR EACH BRICK.

cc. 1-10 X(1): GLOBAL X-COORDINATE OF BRICK ELEMENT NODE CORRESPONDING TO STANDARD BRICK NODE NUMBER ONE (FT).

cc. 11-20 Y(1): GLOBAL Y-COORDINATE OF SAID NODE (FT).

cc. 21-30 DELX: X-DIMENSION OF BRICK (FT).

cc. 31-40 DELY: Y-DIMENSION OF BRICK (FT).

cc. 41-50 XKX: THERMAL CONDUCTIVITY IN X-DIRECTION (BTU/HR.FT-^{o}F)

cc. 51-60 XKY: THERMAL CONDUCTIVITY IN Y-DIRECTION(BTU/HR.FT-^{o}F)

cc. 61-70 THER: PRODUCT OF ELEMENT'S DENSITY(LBM/FT^{3}) AND SPECIFIC HEAT (BTU/LBM-^{o}F)

(NOTE: REPEAT THIS PROCEDURE FOR EACH BRICK ELEMENT.)

SIXTH DATA CARD SET (7I6) ITETRA (I,J)

(OMIT IF THERE ARE NO TRIANGLE ELEMENTS; I.E. IF NTETRA = 0)

INPUT TRIANGLE NODAL SEQUENCE AND TRIANGLE ELEMENT NUMBER.

NOTE: NODAL INPUT MUST BE IN "STANDARD NODAL SEQUENCE."

cc. 1-36 SIX NODES OF SEQUENCE

cc. 37-42 TRIANGLE ELEMENT NUMBER

(NOTE: REPEAT THIS PROCEDURE FOR EACH ELEMENT.)

SEVENTH DATA CARD SET (6F10.4, 3F10.4) TETRA (I,J)

(OMIT IF THERE ARE NO TRIANGLE ELEMENTS, I.E. IF NTETRA=0)

INPUT TRIANGLE PARAMETER DATA FOR EACH TRIANGLE ELEMENT.

CARD ONE (6F10.4)

cc. 1-60 X(1), Y(1), X(2), Y(2), X(3), Y(3)

CARD TWO (3F10.4)

cc. 1-10 XKX: THERMAL CONDUCTIVITY IN X-DIRECTION (BTU/HR-FT-^{o}F)

cc. 11-20 XKY: THERMAL CONDUCTIVITY IN Y-DIRECTION (BTU/HR-FT-^{o}F)

cc. 21-30 THER: PRODUCT OF DENSITY (LBM/FT^3) AND SPECIFIC HEAT (BTU/LBM-^{o}F)

(NOTE: REPEAT FOR EACH TRIANGLE ELEMENT)

EIGHTH DATA CARD SET (6F11.5) BEGIN (I)

cc. 1-66 FIRST SIX INITIAL TEMPERATURES. (^{o}F). INPUT MUST BE IN NODAL ORDER OF MAGNITUDE. INPUT ALL INITIAL TEMPERATURES, (INCLUDING BOUNDARY CONDITIONS), FOR I = 1 TO NNODES.

FOURTH DATA CARD SET (12I6,/,12I6,/3I6) IBRICK (I,J)

(OMIT IF THERE ARE NO BRICK ELEMENTS, I.E. NBRICK = 0)

INPUT BRICK NODAL SEQUENCE, IN ELEMENT ORDER OF MAGNITUDE.

NOTE: BRICK NODAL INPUT MUST BE IN "STANDARD NODAL SEQUENCE".

DATA INPUT IS PUNCHED ON THREE CARDS FOR EACH BRICK ELEMENT INTO THE MATRIX IBRICK (I,J) AS FOLLOWS:

CARD ONE: (12I6)

cc. 1-72 FIRST TWELVE NODES IN SEQUENCE.

CARD TWO: (12I6)

cc. 1-72 NEXT TWELVE NODES IN SEQUENCE.

CARD THREE: (3I6)

cc. 1-12 LAST TWO NODES IN SEQUENCE

cc. 13-18 ELEMENT NUMBER

(NOTE: REPEAT THIS PROCEDURE FOR EACH BRICK ELEMENT.)

FIFTH DATA CARD SET (7F10.4, 6F10.4) BRICK (I,J)

(OMIT IF THERE ARE NO BRICK ELEMENTS, I.E. NBRICK = 0)

INPUT BRICK PARAMETER DATA FOR EACH BRICK AS FOLLOWS:

CARD ONE: (7F10.4)

cc. 1-10 X(1): GLOBAL X-COORDINATE OF BRICK ELEMENT NODE CORRESPONDING TO STANDARD BRICK NODE NUMBER ONE(FT).

cc. 11-20 Y(1): GLOBAL Y-COORDINATE OF SAID NODE (FT).

cc. 21-30 Z(1): GLOBAL Z-COORDINATE OF SAID NODE (FT).

cc. 31-40 DELX: X-DIMENSION OF BRICK (FT).

cc. 41-50 DELY: Y-DIMENSION OF BRICK (FT.)

cc. 51-60 DELZ: Z-DIMENSION OF BRICK (FT).

cc. 61-70 FROZEN THERMAL CONDUCTIVITY(BTU/HR.FT-^{o}F)

FIRST CONTROL CARD (I2)

cc. 2	KOSE: DIMENSION OF PROBLEM (2)

SECOND CONTROL CARD (I1)

cc. 1	KODE 1: "0" IS FOR ISOTROPIC HEAT CONDUCTION MODEL WITH ISOTHERMAL PHASE CHANGE.

FIRST DATA CARD SET (6I4, 2F10.5)

cc. 1-4	NBAND: BANDWIDTH OF GLOBAL MATRICES
cc. 5-8	NNODES: NUMBER OF NODES IN PROBLEM
cc. 9-12	NELE: TOTAL NUMBER OF ELEMENTS
cc. 13-16	NBRICK: NUMBER OF BRICK ELEMENTS
cc. 17-20	NTETRA: NUMBER OF TETRAHEDRON ELEMENTS
cc. 21-24	NUMBC: NUMBER OF NODES WITH SPECIFIED TEMPERATURES
cc. 25-34	THETA: TIME STEP, EXPRESSED IN HOURS
cc. 35-44	DAYS: DURATION OF TEST, EXPRESSED IN HOURS

SECOND DATA CARD SET (6F11.5)

cc. 1-66	BC(I): TEMPERATURES OF BOUNDARY CONDITIONS, EXPRESSED IN ^{o}F., READ IN NUMERICAL ORDER-OF-MAGNITUDE OF CORRESPONDING GLOBAL NODE NUMBERS.

THIRD DATA CARD SET (11I6)

cc. 1-66	NBC(I): GLOBAL NODE NUMBERS OF BOUNDARY CONDITION, READ IN NUMERICAL ORDER-OF-MAGNITUDE.

CARD TWO (6F10.4)

cc. 1-10 UNFROZEN THERMAL CONDUCTIVITY

cc. 11-20 FROZEN HEAT CAPACITY

cc. 21-30 UNFROZEN HEAT CAPACITY

cc. 31-40 FROZEN ELEMENT DENSITY

cc. 41-50 UNFROZEN ELEMENT DENSITY.

cc. 51-60 LATENT HEAT PER UNIT WEIGHT.

(NOTE: REPEAT THIS PROCEDURE FOR EACH BRICK ELEMENT.)

SIXTH DATA CARD SET (11I6) ITETRA (I,J)

(OMIT IF THERE ARE NO TETRAHEDRON ELEMENTS: I.E., IF NTETRA=0)

INPUT TETRAHEDRON NODAL SEQUENCE AND TETRAHEDRON ELEMENT NUMBER. NOTE: NODAL INPUT MUST BE IN "STANDARD NODAL SEQUENCE"

CARD ONE: (11I6)

cc. 1-60 TEN NODES IN ABOVE SEQUENCE

cc. 61-66 ELEMENT NUMBER.

(NOTE: REPEAT THIS PROCEDURE FOR EACH TETRAHEDRON ELEMENT.)

SEVENTH DATA CARD SET (6F10.4, 6F10.4, 6F10.4) TETRA (I,J)

(OMIT IF THERE ARE NO TETRAHEDRON ELEMENTS, I.E., IF NTETRA=0)

INPUT TETRAHEDRON PARAMETER DATA FOR EACH TETRAHEDRON ELEMENT.

CARD ONE (6F10.4)

cc. 1-60 X(1), Y(1), Z(1), X(2), Y(2), Z(2)

CARD TWO (6F10.4)

cc. 1-60 X(3), Y(3), Z(3), X(4), Y(4), Z(4)

CARD THREE (7F10.4)

cc. 1-10 FROZEN ELEMENT CONDUCTIVITY (BTU/HR-FT-^{o}F)

cc. 11-20 UNFROZEN ELEMENT CONDUCTIVITY

cc. 21-30 FROZEN HEAT CAPACITY

cc. 31-40 UNFROZEN HEAT CAPACITY

cc. 41-50 FROZEN ELEMENT DENSITY

cc. 51-60 UNFROZEN ELEMENT DENSITY

cc. 61-70 LATENT HEAT PER UNIT WEIGHT

(NOTE: REPEAT FOR EACH TETRAHEDRON ELEMENT)

EIGHTH DATA CARD SET (6F11.5) BEGIN (I)

cc. 1-66 FIRST SIX INITIAL TEMPERATURES (^{o}F). INPUT MUST BE IN NODAL ORDER OF MAGNITUDE. INPUT ALL INITIAL TEMPERATURES, (INCLUDING BOUNDARY CONDITIONS), FOR I=1 TO NNODES.

NINTH DATA CARD SET (35F2.0)

INPUT INITIAL PHASE DATA: "1" MEANS ELEMENT IS THAWED
"0" MEANS ELEMENT IS FROZEN

cc. 1-70 35 ELEMENT PHASE INDICATIONS

(REPEAT ABOVE PROCEDURE AS NEEDED)

SECTION 4

FIRST CONTROL CARD (I2)

cc. 2 KODE: DIMENSION OF PROBLEM (2)

SECOND CONTROL CARD (I1)

cc. 1 KODE 1: "0" IS FOR ANISOTROPIC TRANSIENT HEAT CONDUCTION MODEL WITH ISOTHERMAL PHASE CHANGE

FIRST DATA CARD SET (6F11.5)

cc. 1-4 NBAND: BANDWIDTH OF GLOBAL MATRICES

cc. 5-8 NNODES: NUMBER OF NODES IN PROBLEM

cc. 9-12 NELE: TOTAL NUMBER OF ELEMENTS

cc. 13-16 NBRICK: NUMBER OF BRICK ELEMENTS

cc. 17-20 NTETRA: NUMBER OF TETRAHEDRON ELEMENTS

cc. 21-24 NUMBC: NUMBER OF NODES WITH SPECIFIED TEMPERATURES

cc. 25-34 THETA: TIME STEP, EXPRESSED IN HOURS

cc. 35-44 DAYS: DURATION OF TEST, EXPRESSED IN HOURS

SECOND DATA CARD SET (6F11.5)

cc. 1-66 BC(I): TEMPERATURES OF BOUNDARY CONDITIONS, EXPRESSED IN ^{O}F., READ IN NUMBERICAL ORDER-OF-MAGNITUDE OF CORRESPONDING GLOBAL NODE NUMBERS.

THIRD DATA CARD SET (11I6)

cc. 1-66 NBC(I): GLOBAL NODE NUMBERS OF BOUNDARY CONDITION, READ IN NUMERICAL ORDER-OF-MAGNITUDE.

FOURTH DATA CARD SET (10I6) IBRICK (I,J)

(OMIT IF THERE ARE NO BRICK ELEMENTS, I.E. NBRICK=0)

INPUT BRICK NODAL SEQUENCE, IN ELEMENT ORDER OF MAGNITUDE

NOTE: BRICK NODAL INPUT MUST BE IN "STANDARD NODAL SEQUENCE".

cc. 1-54 NINE NODES OF SEQUENCE

cc. 55-60 ELEMENT NUMBER

FIFTH DATA CARD SET (6F10.4, 5F10.4) BRICK (I,J)

(OMIT IF THERE ARE NO BRICK ELEMENTS: I.E. IF NBRICK=0)

INPUT BRICK PARAMETER DATA FOR EACH BRICK ELEMENT.

CARD ONE (6F10.4)

cc. 1-10 X(1): GLOBAL X-COORDINATE OF BRICK ELEMENT NODE CORRESPONDING TO STANDARD BRICK NODE NUMBER ONE (FT)

cc. 11-20 Y(1): GLOBAL Y-COORDINATE OF SAID NODE (FT).

cc. 21-30 DELX: X-DIMENSION OF BRICK (FT).

cc. 31-40 DELY: Y-DIMENSION OF BRICK (FT)

cc. 41-50 FROZEN THERMAL CONDUCTIVITY (BTU/HR-FT-^{o}F)

cc. 51-60 UNFROZEN THERMAL CONDUCTIVITY

CARD TWO (5F10.4)

cc. 1-10 FROZEN HEAT CAPACITY

cc. 11-20 UNFROZEN HEAT CAPACITY

cc. 21-30 FROZEN ELEMENT DENSITY

cc. 31-40 UNFROZEN ELEMENT DENSITY

cc. 41-50 LATENT HEAT PER UNIT WEIGHT

(NOTE: REPEAT THIS PROCEDURE FOR EACH BRICK ELEMENT)

SIXTH DATA CARD SET (7I6) ITETRA (I,J)

(OMIT IF THERE ARE NO TRIANGLE ELEMENTS: I.E. IF NTETRA=0)

INPUT TRIANGLE NODAL SEQUENCE AND TRIANGLE ELEMENT NUMBER.

NOTE: NODAL INPUT MUST BE IN "STANDARD NODAL SEQUENCE."

cc. 1-36 SIX NODES OF SEQUENCE

cc. 37-42 TRIANGLE ELEMENT NUMBER

NOTE: REPEAT THIS PROCEDURE FOR EACH ELEMENT.)

SEVENTH DATA CARD SET (7F10.4, 6F10.4)

(OMIT IF THERE ARE NO THIANGLE ELEMENTS: I.E. IF NTETRA=0).

INPUT TRIANGLE PARAMETER DATA FOR EACH TRIANGLE.

CARD ONE (7F10.4)

cc. 1-60 X(1), Y(1), X(2), Y(2), X(3), Y(3)

cc. 61-70 FROZEN THERMAL CONDUCTIVITY (BTU/HR-FT-^{o}F)

CARD TWO (6F10.4)

cc. 1-10 UNFROZEN THERMAL CONDUCTIVITY

cc. 11-20 FROZEN HEAT CAPACITY

cc. 21-30 UNFROZEN HEAT CAPACITY

cc. 31-40 FROZEN ELEMENT DENSITY

cc. 41-50 UNFROZEN ELEMENT DENSITY

cc. 51-60 LATENT HEAT PER UNIT WEIGHT

(NOTE: REPEAT FOR EACH TRIANGLE ELEMENT)

EIGHTH DATA CARD SET (6F11.5) BEGIN (I)

cc. 1-66 FIRST SIX INITIAL TEMPERATURES. (^{o}F). INPUT MUST BE IN NODAL ORDER OF MAGNITUDE. INPUT ALL INITIAL TEMPERATURES, (INCLUDING BOUNDARY CONDITIONS), FOR I = 1 TO NNODES.

NINTH DATA CARD SET (35F2.0)

INPUT INITIAL PHASE DATA: "1" MEANS ELEMENT IS THAWED

"0" MEANS ELEMENT IS FROZEN

cc. 1-70 35 ELEMENT PHASE INDICATIONS

(REPEAT ABOVE PROCEDURE AS NEEDED)

SECTION 5

ELEMENT NUMBERING PATTERN (2-D)

Triangle elements are to be numbered from 1 to NTETRA. The BRICK elements are to be numbered in increments of "2" starting from (NTETRA + 2) to ((NTETRA + 2) + 2(NBRICK)). This is because there are two triangle elements in each brick element.

ELEMENT NUMBERING PATTERN (3-D)

Tetrahedron elements are to be numbered form 1 to NTETRA. The BRICK elements are to be numbered in increments of "5" starting from (NTETRA + 5) to ((NTETRA + 5) + 5 (NBRICK)). This is due to each BRICK element being composed of 5 tetrahedron elements.

STANDARD NODAL SEQUENCES (2 or 3-D)

Refer to Chptrs. 2 and 3 for illustrations of the nodal numbering input schemes.

```
C---
C---
C--- THIS PROGRAM SOLVES A TWO-D OR THREE-D ANISOTROPIC TRANSIENT HEAT
C--- PROBLEM. A FINITE ELEMENT VARIATIONAL TECHNIQUE IS USED. THE BASIC
C--- DISCRETIZATION ELEMENTS ARE TRIANGLES (2-D) AND TETRAHEDRA (FOR
C--- THREE-D). VARIABLE ELEMENT DIMENSIONS ARE ASSUMED AND A QUADRATIC
C--- POLYNOMIAL SHAPE FUNCTION IS USED. THE TIME DOMAIN SOLUTION IS BY
C--- THE CRANK-NICOLSON METHOD. ISOTHERMAL PHASE CHANGE IS APPROXIMATED
C--- FOR AN ISOTROPIC CONTINUUM.
C---
C---
C--- FORMAT STATEMENTS
C---
    1 FORMAT(6I4,2F10.5)
    2 FORMAT(6F11.5)
    3 FORMAT(11I6)
    4 FORMAT(2X,'**TRIANGLE NODE INPUT SEQUENCE**',/,2X,6I6,/)
    5 FORMAT(2X,'**TRIANGLE CORNER COORDINATES IN NODAL SEQUENCE**',/,3(
     12X,'X =',F10.4,20X,'Y =',F10.4,/),2X,'XKX =',F10.4,15X,'XKY =',F10
     1.4,15X,'THER =',F10.4,/)
    6 FORMAT(6F11.5)
    7 FORMAT(I1)
    8 FORMAT(7F10.4,/,6F10.4)
    9 FORMAT(6F10.4,/,5F10.4)
   10 FORMAT(3X,'TIME EQUALS',F8.2,/,3X,'COUNT EQUALS',F8.2,//)
   11 FORMAT(5X,'NODE',I5,2X,'TEMP',F10.4)
   12 FORMAT(/,38X,'PROGRAM INPUT PARAMETER DATA',/)
   13 FORMAT(5X,'NBAND =',I4,14X,'NNODES =',I4,13X,'NELE =',I4,15X,'NBRI
     1CK =',I4,/,5X,'NTETRA =',I4,13X,'NUMBC =',I4,14X,'THETA =',F6.2,12
     1X,'DAYS =',F8.2,//)
   14 FORMAT(/,30X,'SPECIFIED TEMPERATURE BOUNDARY CONDITION INPUT',/)
   15 FORMAT(7(3X,'NODE',5X,'TEMP.'))
   16 FORMAT(7(2X,I4,5X,F8.2))
   17 FORMAT(/,42X,'BRICK DATA BLOCK INPUT',/)
   18 FORMAT(5(2X,'NODE',I4,1X,'IS PHASE',F2.0,2X))
   19 FORMAT(/,39X,'TETRAHEDRON DATA BLOCK INPUT',/)
   20 FORMAT(2X,8('**************'))
   21 FORMAT(2X,8('--------------'))
   22 FORMAT(3X,'STANDARD NODE NUMBER ONE COORDINATES',/,3X,'X =',F10.5,
     13X,'Y =',F10.5,'Z =',F10.5,/,3X,'DELX =',F10.5,3X,'DELY =',F10.5,
     13X,'DELZ =',F10.5,/,3X,'FROZEN THERMAL CONDUCTIVITY IS',F10.5,10X,
     1'UNFROZEN THERMAL CONDUCTIVITY IS',F10.5,/,3X,'FROZEN HEAT CAPACIT
     1Y IS',F10.5,10X,'UNFROZEN HEAT CAPACITY IS',F10.5,/,3X,'FROZEN ELE
     1MENT DENSITY IS',F10.5,10X,'UNFROZEN ELEMENT DENSITY IS',F10.5,/,3
     1X,'LATENT HEAT PER UNIT WEIGHT IS',F10.5,/)
   23 FORMAT(/,29X,'INITIAL TEMPERATURES IN NODAL ORDER OF MAGNITUDE',/)
   24 FORMAT(3X,'STANDARD NODE NUMBER ONE COORDINATES',/,3X,'X =',F10.5,
     1'Y=',F10.5,/,3X,'DELX=',F10.5,'DELY=',F10.5,/,3X,'FROZEN THERMAL C
     1ONDUCTIVITY IS',F10.5,10X,'UNFROZEN THERMAL CONDUCTIVITY IS',F10.5
     1,/,3X,'FROZEN HEAT CAPACITY IS',F10.5,10X,'UNFROZEN HEAT CAPACITY
     1IS',F10.5,/,3X,'FROZEN ELEMENT DENSITY IS',F10.5,10X,'UNFROZEN ELE
     1MENT DENSITY IS',F10.5,/,3X,'LATENT HEAT PER UNIT WEIGHT IS',F10.5
     1,/)
   25 FORMAT(6F10.4,/,6F10.4,/,7F10.4)
   27 FORMAT(3X,'TETRAHEDRON CORNER COORDINATES IN STANDARD NODAL SEQUEN
     1CE',/,4(2X,'X =',F10.5,10X,'Y =',F10.5,10X,'Z =',F10.5,/),3X,'FROZ
```

```
     1EN THERMAL CONDUCTIVITY IS',F10.5,10X,'UNFROZEN THERMAL CONDUCTIVI
     1TY IS',F10.5,/,3X,'FROZEN HEAT CAPACITY IS',F10.5,10X,'UNFROZEN HE
     1AT CAPACITY IS',F10.5,/,3X,'FROZEN ELEMENT DENSITY IS',F10.5,10X,'
     1UNFROZEN ELEMENT DENSITY IS',F10.5,/,3X,'LATENT HEAT PER UNIT WEIG
     1HT IS',F10.5,/)
   28 FORMAT(3X,'TRIANGLE CORNER COORDINATES IN STANDARD NODAL SEQUENCE'
     1,/,3(2X,'X =',F10.5,10X,'Y =',F10.5,/),3X,'FROZEN THERMAL CONDUCTI
     1VITY IS',F10.5,10X,'UNFROZEN THERMAL CONDUCTIVITY IS',F10.5,/,3X,'
     1FROZEN HEAT CAPACITY IS',F10.5,10X,'UNFROZEN HEAT CAPACITY IS',F10
     1.5,/,3X,'FROZEN ELEMENT DENSITY IS',F10.5,10X,'UNFROZEN ELEMENT DE
     1NSITY IS',F10.5,/,3X,'LATENT HEAT PER UNIT WEIGHT IS',F10.5,/)
   29 FORMAT(3X,'NODAL PHASE STATUS',/)
   30 FORMAT(12I6,/,12I6,/,3I6)
   31 FORMAT(6F10.4,/,4F10.4)
   32 FORMAT(2X,'**ELEMENT NUMBER**',I4,/)
   33 FORMAT(2X,'**BRICK NODE INPUT SEQUENCE**',/,2X,26I4,/)
   34 FORMAT(2X,'**STANDARD NODE NUMBER-ONE COORDINATES**',/,2X,'X =',F1
     10.4,20X,'Y =',F10.4,20X,'Z =',F10.4,/,/,2X,'DELX =',F10.4,20X,'DEL
     1Y =',F10.4,20X,'DELZ =',F10.4,/,/,2X,'XKX =',F10.4,15X,'XKY =',F10
     1.4,15X,'XKZ =',F10.4,15X,'THER =',F10.4,/)
   35 FORMAT(11I6)
   36 FORMAT(6F10.4,/,6F10.4,/,4F10.4)
   37 FORMAT(2X,'**TETRAHEDRON NODE INPUT SEQUENCE**',/,2X,10I6,/)
   38 FORMAT(2X,'**TETRAHEDRON CORNER COORDINATES IN NODAL SEQUENCE**',/
     1,4(2X,'X =',F10.4,20X,'Y =',F10.4,20X,'Z =',F10.4,/),2X,'XKX =',F1
     10.4,15X,'XKY =',F10.4,15X,'XKZ =',F10.4,15X,'THER =',F10.4,/)
   39 FORMAT(35F2.0)
   40 FORMAT(I2)
   41 FORMAT(2X,'*',/,2X,'**',/,2X,'***')
   42 FORMAT(2X,'****',2X,'DIMENSION OF PROBLEM IS',I2)
   43 FORMAT(2X,'***',/,2X,'**',/,2X,'*')
   44 FORMAT(10I5)
   45 FORMAT(7F10.4)
   46 FORMAT(2X,'**BRICK NODE INPUT SEQUENCE**',/,2X,9I5,/)
   47 FORMAT(2X,'**STANDARD NODE NUMBER ONE COORDINATES**',/,2X,'X =',F1
     10.4,20X,'Y =',F10.4,/,2X,'DELX =',F10.4,20X,'DELY =',F10.4,/,/,2X,
     1'XKX =',F10.4,15X,'XKY =',F10.4,15X,'THER =',F10.4,/)
   48 FORMAT(7I6)
   49 FORMAT(6F10.4,/,3F10.4)
      COMMON/BLK 1/GK(100,40)
      COMMON/BLK 2/GC(100,40)
      COMMON/BLK 3/ZZ(100)
      COMMON/BLK 4/ST(100,40)
      COMMON/BLK 5/WEIGHT(100)
      COMMON/BLK 6/MATRIX(100,5)
      COMMON/BLK 7/ HEAT(100)
      COMMON/BLK 8/ CO(100)
      COMMON/ BLK 9/ CP(100)
      COMMON/BLK 10/ SAVE(100,40)
      DIMENSION BC(50),NBC(50),TETRA(5,19),BRICK(7,13),CORD(8,3),
     1X(4),Y(4),Z(4),NOD(10),IT(100),BEGIN(100),ITETRA(5,11),IBRICK(5,27
     1),ACCUM(100),NS(5,10),MIHAWD(40)
C*
C**
C***
C**********SEGMENT-ONE-----DATA INPUT**********
```

```
C***
C**
C*
C-----
C----- DEFINITIONS OF COMPUTER VARIABLE CODING:
C----- GK=GLOBAL "K" MATRIX
C----- GC=GLOBAL "C" MATRIX
C----- BC= BOUNDARY CONDITION VECTOR
C----- NBC= NODE NUMBERS OF BOUNDARY CONDITIONS
C----- BEGIN= INITIAL CONDITION VECTOR
C----- TETRA= TRIANGLE DATA BLOCK (2-DIMENSION PROBLEM)
C----- TETRA= TETRAHEDRON DATA BLOCK (3-DIMENSION PROBLEM)
C----- ITETRA= TRIANGLE INTEGER INPUT(2-DIMENSION PROBLEM)
C----- ITETRA= TETRAHEDRON INTEGER INPUT(3-DIMENSION PROBLEM)
C----- BRICK = BRICK DATA BLOCK
C----- CORD= BRICK CORNER COORDINATES
C----- IBRICK= BRICK INTEGER INPUT
C----- X(3),Y(3)=TRIANGLE ELEMENT CORNER COORDINATES
C----- X(4),Y(4),Z(4)= TETRAHEDRON ELEMENT CORNER COORDINATES
C----- NOD= TRIANGLE ELEMENT NODAL SEQUENCE
C----- NOD=TETRAHEDRON ELEMENT NODAL SEQUENCE
C----- TT = TEMPORARY VECTOR OF SPECIFIED BOUNDARY TEMPERATURES
C-----
C----- DEFINE DIMENSION OF PROBLEM
C----- KODE= 2 IMPLIES A TWO-D PROBLEM
C----- KODE=3 IMPLIES A THREE-D PROBLEM
      READ(1,40) KODE
      WRITE(3,20)
      WRITE(3,41)
      WRITE(3,42) KODE
      WRITE(3,43)
      WRITE(3,20)
C-----
C-----
C----- KODE1=0 IMPLIES A PHASE CHANGE TYPE MODEL.
C----- KODE1=2 IMPLIES A STANDARD TRANSIENT HEAT CONDUCTION MODEL.
C----- KODE3=0 IMPLIES NO ELEMENT PHASE UPDATE IN PROGRESS
C----- KODE3=1 IMPLIES ELEMENT PHASE UPDATE IN PROGRESS
      KODE3=0
      READ(1,7) KODE1
      WRITE(3,20)
      WRITE(3,41)
      IF(KODE1.EQ.0) WRITE(3,2503)
      IF(KODE1.EQ.2) WRITE(3,2504)
      WRITE(3,43)
      WRITE(3,20)
C-----
C-----
C----- INPUT PROBLEM PARAMETERS AS DEFINED ABOVE
C-----
      READ(1,1)NBAND,NNODES,NELE,NBRICK,NTETRA,NUMBC,THETA,DAYS
      WRITE(3,29)
      WRITE(3,12)
      WRITE(3,21)
      WRITE(3,13)NBAND,NNODES,NELE,NBRICK,NTETRA,NUMBC,THETA,DAYS
      WRITE(3,20)
```

```
C---
C---
C---   NBAND= BANDWIDTH OF GLOBAL MATRICES
C---   NNODES= NUMBER OF NODES IN PROBLEM
C---   NELE= TOTAL NUMBER OF ELEMENTS
C---   NBRICKS= NUMBER OF BRICK ELEMENTS
C----- NTETRA= NUMBER OF TRIANGLE ELEMENTS
C---   NTETRA= NUMBER OF TETRAHEDRON ELEMENTS
C---   THETA= TIME INCREMENT
C---   DAYS= SIMULATION DURATION
C---   NUMBC= NUMBER OF NODES GIVEN SPECIFIED TEMPERATURES
C---
C---
C---  INPUT BOUNDARY CONDITION DATA
C---
      READ (1,2) (BC(I), I=1,NUMBC)
      READ(1,3)(NBC(I),I=1,NUMBC)
      WRITE(3,20)
      WRITE(3,14)
      WRITE(3,21)
      WRITE(3,15)
      WRITE(3,16)(NBC(I),BC(I),I=1,NUMBC)
      WRITE(3,20)
C---
C---  INPUT "BRICK" ELEMENT DATA
C---
C---  NOTE: A NEGATIVE LATENT HEAT IS A FLAG THAT THE ELEMENT IS FROZEN.
C---  A POSITIVE LATENT HEAT IMPLIES THAT THE ELEMENT IS UNFROZEN.
C---
      IF(NBRICK.EQ.0) GO TO 70
      IF(KODE.EQ.2) GO TO 54
      READ(1,30)((IBRICK(I,J),J=1,27),I=1,NBRICK)
      IF(KODE1.EQ.0) GO TO 51
      READ(1,31)((BRICK(I,J),J=1,10),I=1,NBRICK)
      GO TO 56
   51 READ(1,8)((BRICK(I,J),J=1,13),I=1,NBRICK)
      GO TO 56
   54 READ(1,44)((IBRICK(I,J),J=1,10),I=1,NBRICK)
      IF(KODE1.EQ.0) GO TO 55
      READ(1,45)((BRICK(I,J),J=1,7),I=1,NBRICK)
      GO TO 56
   55 READ(1,9)((BRICK(I,J),J=1,11),I=1,NBRICK)
   56 CONTINUE
      WRITE(3,20)
      WRITE(3,17)
      WRITE(3,21)
      IF(KODE1.EQ.0) GO TO 61
      DO 60 I=1,NBRICK
      IF(KODE.EQ.2) GO TO 57
      WRITE(3,32) IBRICK(I,27)
      WRITE(3,33)(IBRICK(I,J),J=1,26)
      WRITE(3,34)(BRICK(I,J),J=1,10)
      GO TO 58
   57 WRITE(3,32)IBRICK(I,10)
      WRITE(3,46)(IBRICK(I,J),J=1,9)
      WRITE(3,47)(BRICK(I,J),J=1,7)
```

```
   58 CONTINUE
      WRITE(3,20)
   60 CONTINUE
      GO TO 70
   61 DO 69 I=1,NBRICK
      IF(KODE.EQ.2) GO TO 67
      WRITE(3,32) IBRICK(I,27)
      WRITE(3,33)(IBRICK(I,J),J=1,26)
      WRITE(3,22)(BRICK(I,J),J=1,13)
      GO TO 68
   67 WRITE(3,32) IBRICK(I,10)
      WRITE(3,46)(IBRICK(I,J),J=1,9)
      WRITE(3,24)(BRICK(I,J),J=1,11)
   68 CONTINUE
      WRITE(3,20)
   69 CONTINUE
   70 CONTINUE
C---
C--- INPUT TRIANGLE/TETRAHEDRON ELEMENT DATA
C---
      IF(NTETRA.EQ.0) GO TO 97
      IF(KODE.EQ.2) GO TO 72
      READ(1,35)((ITETRA(I,J),J=1,11),I=1,NTETRA)
      IF(KODE1.EQ.0) GO TO 71
      READ(1,36)((TETRA(I,J),J=1,16),I=1,NTETRA)
      GO TO 75
   71 READ(1,25)((TETRA(I,J),J=1,19),I=1,NTETRA)
      GO TO 75
   72 READ(1,48)((ITETRA(I,J),J=1,7),I=1,NTETRA)
      IF(KODE1.EQ.0) GO TO 73
      READ(1,49)((TETRA(I,J),J=1,9),I=1,NTETRA)
      GO TO 75
   73 READ(1,8)((TETRA(I,J),J=1,13),I=1,NTETRA)
   75 CONTINUE
      WRITE(3,20)
      WRITE(3,19)
      WRITE(3,21)
      IF(KODE1.EQ.0) GO TO 91
      DO 88 I=1,NTETRA
      IF(KODE.EQ.2) GO TO 83
      WRITE(3,32)ITETRA(I,11)
      WRITE(3,37)(ITETRA(I,J),J=1,10)
      WRITE(3,38)(TETRA(I,J),J=1,16)
      GO TO 84
   83 WRITE(3,32) ITETRA(I,7)
      WRITE(3,4)(ITETRA(I,J),J=1,6)
      WRITE(3,5)(TETRA(I,J),J=1,9)
   84 CONTINUE
      WRITE(3,20)
   88 CONTINUE
   90 CONTINUE
      GO TO 97
   91 DO 94 I=1,NTETRA
      IF(KODE.EQ.2) GO TO 93
      WRITE(3,32) ITETRA(I,11)
      WRITE(3,37)(ITETRA(I,J),J=1,10)
```

```
      WRITE(3,27)(TETRA(I,J),J=1,19)
      GO TO 94
   93 WRITE(3,32) (TETRA(I,7)
      WRITE(3,4)(TETRA(I,J),J=1,6)
      WRITE(3,28)(TETRA(I,J),J=1,13)
   94 CONTINUE
   97 CONTINUE
C---
C--- INPUT INITIAL CONDITION DATA
C---
      READ (1,6) (BEGIN (I), I= 1, NNODES)
      WRITE(3,20)
      WRITE(3,23)
      WRITE(3,21)
      WRITE(3,15)
      WRITE(3,16)(I,BEGIN(I),I=1,NNODES)
      WRITE(3,20)
      IF(KODE1.EQ.2) GO TO 99
C---
C--- INPUT PHASE STATUS NODAL DATA
C---
C--- ACCUM(I)=1.0 IMPLIES THAT NODE IS INITIALLY UNFROZEN
C--- ACCUM(I)=0.0 IMPLIES THAT NODE IS INITIALLY FROZEN
      READ(1,39)(ACCUM(I),I=1,NNODES)
      WRITE(3,20)
      WRITE(3,29)
      WRITE(3,21)
      WRITE(3,18)(I,ACCUM(I),I=1,NNODES)
      WRITE(3,20)
   99 CONTINUE
C*
C**
C***
C**********SEGMENT-TWO-----PRELIMINARY CALCULATIONS**********
C***
C**
C*
C---
C---  SET GLOBAL MATRICES"GK" AND "GC" EQUAL TO ZERO
C---
      DO 100 I=1,NNODES
      DO 100 J=1,NBAND
      GK (I,J)= 0.0
      GC(I,J)=0.0
  100 CONTINUE
C---
C--- KODE1 = 1 IMPLIES THAT PROGRAM IS IN TIME ADVANCEMENT ROUTINES
C---
      IF(KODE1.EQ.2) GO TO 106
C--- INITIALIZE PHASE CHANGE THERMAL PARAMETERS TO ZERO
C--- HEAT(I)=LATENT HEAT OF FUSION FOR NODE "I"
C--- CU(I)=WEIGHTED THAWED SPECIFIC HEAT FOR NODE "I"
C--- CF(I)=WEIGHTED FROZEN SPECIFIC HEAT FOR NODE "I"
C--- WEIGHT(I)=WEIGHT OF MATERIAL ASSOCIATED TO NODE "I"
C--- MATRIX(I,J)=POINTER ARRAY OF ASSOCIATED ELEMENTS "J" TO NODE "I"
      DO 102 I=1,100
```

```
      HEAT(I)=0.0
      CO(I)=0.0
      CF(I)=0.0
  102 WEIGHT(I)=0.0
      DO 105 I=1,NNODES
  105 MATRIX(I,1)=2
  106 CONTINUE
C---
C--- PREPARE NODAL SEQUENCE POINTER ARRAY
C---
      IF(KODE.EQ.3) GO TO 120
      NS(1,1)=1
      NS(1,2)=3
      NS(1,3)=7
      NS(1,4)=2
      NS(1,5)=9
      NS(1,6)=8
      NS(2,1)=3
      NS(2,2)=5
      NS(2,3)=7
      NS(2,4)=4
      NS(2,5)=6
      NS(2,6)=9
      GO TO 130
  120 NS(1,1)=1
      NS(1,2)=20
      NS(1,3)=18
      NS(1,4)=24
      NS(1,5)=11
      NS(1,6)=10
      NS(1,7)=17
      NS(1,8)=19
      NS(1,9)=25
      NS(1,10)=26
      NS(2,1)=5
      NS(2,2)=20
      NS(2,3)=24
      NS(2,4)=22
      NS(2,5)=13
      NS(2,6)=15
      NS(2,7)=14
      NS(2,8)=26
      NS(2,9)=23
      NS(2,10)=21
      NS(3,1)=5
      NS(3,2)=3
      NS(3,3)=1
      NS(3,4)=20
      NS(3,5)=4
      NS(3,6)=9
      NS(3,7)=13
      NS(3,8)=2
      NS(3,9)=11
      NS(3,10)=12
      NS(4,1)=5
      NS(4,2)=1
```

```
      NS(4,3)=7
      NS(4,4)=24
      NS(4,5)=9
      NS(4,6)=6
      NS(4,7)=15
      NS(4,8)=8
      NS(4,9)=16
      NS(4,10)=17
      NS(5,1)=5
      NS(5,2)=1
      NS(5,3)=24
      NS(5,4)=20
      NS(5,5)=9
      NS(5,6)=15
      NS(5,7)=13
      NS(5,8)=17
      NS(5,9)=26
      NS(5,10)=11
  130 CONTINUE
C---
C--- DEFINE THE FREEZING POINT DEPRESSION "FPD"
C---
      FPD=32.0
C*
C**
C***
C******SEGMENT-THREE---DETERMINE ELEMENT CONDUCTION AND CAPACITANCE
C******MATRICES FOR INCORPORATION INTO GLOBAL MATRICES.
C******FORM GLOBAL CONDUCTION AND CAPACITANCE MATRICES**********
C***
C**
C*
C---
C---  PROCESS BRICK ELEMENTS
C---
C--- PROCESS THREE-DIMENSIONAL BRICK ELEMENTS
C--- BRICK ELEMENTS (FOR THREE-D PROBLEM) ARE DISCRETIZED INTO FIVE
C--- TETRAHEDRON ELEMENTS DENOTED AS "SUB 1" TO "SUB 5". MAIN PROGRAM
C--- PREPARES PARAMETERS FOR EACH TETRAHEDRON TO BE TRANSFERRED TO
C--- SUBROUTINE SETUP.
C---
C--- DEFINITIONS FOR PHASE CHANGE SIMULATION:
C--- HET=LATENT HEAT PER UNIT WEIGHT
C--- XU=UNFROZEN HEAT CAPACITY              XF=FROZEN HEAT CAPACITY
C--- DENSTY=DENSITY,OR UNIT WEIGHT          THER=HEAT CAPACITY*DENSITY
C---
      KODE2=0
      IF(NBRICK.EQ.0) GO TO 230
      I=0
      IF(KODE.EQ.2) GO TO 205
  135 I=I+1
      NUMBER = IBRICK(I,27)
C---
C--- ASSIGN "BRICK" DIMENSIONS
C---
      XXX=BRICK(I,7)
```

```
      XKY=BRICK(I,8)
      XKZ=BRICK(I,9)
      THER=BRICK(I,10)
      IF(KODE1.EQ.2) GO TO 145
C---- ASSIGN PHASE THERMAL PARAMETERS
      HEI=BRICK(I,13)
C---- TEST IF ELEMENT IS INITIALLY FROZEN OR UNFROZEN
      IF(HEI.LT.0.0) GO TO 140
C---- ELEMENT IS UNFROZEN
      XKX=BRICK(I,8)
      DENSTY=BRICK(I,12)
      XU=BRICK(I,10)
      XF=BRICK(I,9)
      THER=XU*DENSTY
      NTEST=1
      GO TO 143
  140 CONTINUE
C---- ELEMENT IS FROZEN
      XKX=BRICK(I,7)
      DENSTY=BRICK(I,11)
      XF=BRICK(I,9)
      XU=BRICK(I,10)
      THER=XF*DENSTY
      HEI=-HEI
      NTEST=-1
  143 CONTINUE
      XKY=XKX
      XKZ=XKX
  145 CONTINUE
      IF(KODE1.EQ.1) I=K6
      DELX=BRICK(I,4)
      DELY=BRICK(I,5)
      DELZ=BRICK(I,6)
      V=DELX*DELY*DELZ/6.0
C---
C---- DETERMINE "BRICK" COORDINATES
C---
      CORD(1,1)=BRICK(I,1)
      CORD(1,2)=BRICK(I,2)
      CORD(1,3)=BRICK(I,3)
      CORD(2,1) = CORD(1,1) + DELX
      CORD(2,2)= CORD(1,2)
      CORD(2,3)= CORD (1,3)
      CORD(3,1)= CORD(2,1)
      CORD(3,2)=CORD(1,2)+DELY
      CORD(3,3)= CORD(1,3)
      CORD(4,1)= CORD(1,1)
      CORD(4,2)= CORD(3,2)
      CORD(4,3)= CORD(1,3)
      CORD(5,1)= CORD(1,1)
      CORD(5,2)= CORD(1,2)
      CORD(5,3)= CORD(1,3)-DELZ
      CORD(6,1)= CORD(2,1)
      CORD(6,2)=CORD(1,2)
      CORD(6,3)= CORD(5,3)
      CORD(7,1)= CORD(2,1)
```

```
      CORD(7,2)= CORD(3,2)
      CORD(7,3)= CORD(5,3)
      CORD(8,1)= CORD(1,1)
      CORD(8,2)= CORD (3,2)
      CORD(8,3)= CORD(5,3)
      IF(KODE1.NE.1) GO TO 150
      IF(JX-2) 151,154,148
  148 IF(JX-4) 157,160,163
  150 CONTINUE
C---
C---  PROCESS SUB 1
C---
  151 CONTINUE
      X(1)= CORD (1,1)
      Y(1)= CORD (1,2)
      Z(1)= CORD (1,3)
      X(2)= CORD(6,1)
      Y(2)= CORD (6,2)
      Z(2)= CORD (6,3)
      X(3)= CORD (5,1)
      Y(3)= CORD(5,2)
      Z(3)= CORD (5,3)
      X(4)= CORD (8,1)
      Y(4)= CORD(8,2)
      Z(4)= CORD(8,3)
      DO 152 LK=1,10
      LKK=NS(1,LK)
  152 NOD(LK)=IBRICK(I,LKK)
      CALL SETUP(V,XKX,XKY,XKZ,THER,X,Y,Z,NOD,NUMBER,KODE1,DENSTY,HLT,XO
     1,XF)
      IF(KODE1.EQ.1) GO TO 610
      IF(KODE1.EQ.2) GO TO 154
      NBS=0
      DO 153 LK=1,10
      LKK=NS(1,LK)
      NBR=IBRICK(I,LKK)
      IF(ACCUM(NBR).GT.0.0) NBS=NBS+1
  153 CONTINUE
      NBR=NUMBER-4
      WIDTH(NBR)=NBS*WIDEST
C---
C---  PROCESS SUB 2
C---
  154 CONTINUE
      X(1)= CORD (3,  1)
      Y(1)= CORD(3,2)
      Z(1)= CORD( 3,3)
      X(2)= CORD(6,  1)
      Y(2)= CORD (  6,2)
      Z(2)= CORD (  6,3)
      X(3)= CORD (  8,1)
      Y(3)= CORD(  5,2)
      Z(3)= CORD (  8,3)
      X(4)= CORD (  7,1)
      Y(4)= CORD ( 7,2)
      Z(4)= CORD ( 7,3)
```

```
      DO 155 LK=1,10
      LKK=NS(2,LK)
  155 NOD(LK)=IBRICK(I,LKK)
      CALL SETUP(V,XKX,XKY,XKZ,THER,X,Y,Z,NOD,NUMBER,KODE1,DENSTY,HET,XU
     1,XF)
      IF(KODE1.EQ.1) GO TO 810
      IF(KODE1.EQ.2) GO TO 157
      NBS=0
      DO 156 LK=1,10
      LKK=NS(2,LK)
      NBR=IBRICK(I,LKK)
      IF(ACCUM(NBR).GT.0.0)NBS=NBS+1
  156 CONTINUE
      NBR=NUMBER-3
      NTHAWD(NBR)=NBS*NTEST
C---
C---  PROCESS SUB 3
C---
  157 CONTINUE
      X(1)= CORD (3,  1)
      Y(1)= CORD(3,2)
      Z(1)= CORD( 3,3)
      X(2)= CORD ( 2,1)
      Y(2)= CORD (  2,2)
      Z(2)= CORD ( 2,3)
      X(3)= CORD ( 1,1)
      Y(3)= CORD(  1,2)
      Z(3)= CORD ( 1,3)
      X(4)= CORD ( 6,1)
      Y(4)= CORD ( 6,2)
      Z(4)= CORD ( 6,3)
      DO 158 LK=1,10
      LKK=NS(3,LK)
  158 NOD(LK)=IBRICK(I,LKK)
      CALL SETUP(V,XKX,XKY,XKZ,THER,X,Y,Z,NOD,NUMBER,KODE1,DENSTY,HET,XU
     1,XF)
      IF(KODE1.EQ.1) GO TO 810
      IF(KODE1.EQ.2) GO TO 160
      NBS=0
      DO 159 LK=1,10
      LKK=NS(3,LK)
      NBR=IBRICK(I,LKK)
      IF(ACCUM(NBR).GT.0.0) NBS=NBS+1
  159 CONTINUE
      NBR=NUMBER-2
      NTHAWD(NBR)=NBS*NTEST
C---
C---  PROCESS SUB4
C---
  160 CONTINUE
      X(1)= CORD (3,  1)
      Y(1)= CORD(3,2)
      Z(1)= CORD( 3,3)
      X(2)= CORD (1,1)
      Y(2)= CORD ( 1,2)
      Z(2)= CORD ( 1,3)
```

```
      X(3)= CORD ( 4,1)
      Y(3)= CORD(  4,2)
      Z(3)= CORD ( 4,3)
      X(4)= CORD ( 8,1)
      Y(4)= CORD ( 8,2)
      Z(4)= CORD ( 8,3)
      DO 161 LK=1,10
      LKK=NS(4,LK)
  161 NOD(LK)=IBRICK(I,LKK)
      CALL SETUP(V,XKX,XKY,XKZ,THER,X,Y,Z,NOD,NUMBER,KODE1,DENSTY,HET,XU
     1,XF)
      IF(KODE1.EQ.1) GO TO 810
      IF(KODE1.EQ.2) GO TO 163
      NBS=0
      DO 162 LK=1,10
      LKK=NS(4,LK)
      NBR=IBRICK(I,LKK)
      IF(ACCUM(NBR).GT.0.0) NBS=NBS+1
  162 CONTINUE
      NBR=NUMBER-1
      NTHAWD(NBR)=NBS*NTEST
C---
C--- PROCESS SUB 5
C---
  163 CONTINUE
      X(1)= CORD (3,  1)
      Y(1)= CORD(3,2)
      Z(1)= CORD( 3,3)
      X(2)= CORD (1,1)
      Y(2)= CORD ( 1,2)
      Z(2)= CORD ( 1,3)
      X(3)= CORD ( 5,1)
      Y(3)= CORD(  5,2)
      Z(3)= CORD ( 5,3)
      X(4)= CORD (6,1)
      Y(4)= CORD ( 6,2)
      Z(4)= CORD ( 6,3)
      DO 164 LK=1,10
      LKK=NS(5,LK)
  164 NOD(LK)=IBRICK(I,LKK)
      V=V*2.0
      CALL SETUP(V,XKX,XKY,XKZ,THER,X,Y,Z,NOD,NUMBER,KODE1,DENSTY,HET,XU
     1,XF)
      IF(KODE1.EQ.1) GO TO 810
      IF(KODE1.EQ.2) GO TO 202
      NBS=0
      DO 201 LK=1,10
      LKK=NS(5,LK)
      NBR=IBRICK(I,LKK)
      IF(ACCUM(NBR).GT.0.0) NBS=NBS+1
  201 CONTINUE
      NTHAWD(NUMBER)=NBS*NTEST
  202 CONTINUE
      IF(I.GE.NBRICK) GO TO 250
      GO TO 135
C---
```

```
C---
C---PROCESS TWO-DIMENSIONAL BRICK ELEMENTS
C--- BRICK ELEMENTS (FOR TWO-D PROBLEM) ARE DISCRETIZED INTO TWO
C--- TRIANGLE ELEMENTS, DENOTED AS "SUB 1" AND "SUB 2". MAIN PROGRAM
C--- PREPARES PARAMETERS FOR EACH TRIANGLE TO BE TRANSFERRED TO SUBROU-
C--- TINE SETUP2.
C---
C---
  205 I=I+1
      NUMBER=IBRICK(I,10)
C---
C--- ASSIGN "BRICK" DIMENSIONS
C---
      XKX=BRICK(I,5)
      XKY=BRICK(I,6)
      THER=BRICK(I,7)
      IF(KODE1.EQ.2) GO TO 207
C--- ASSIGN PHASE THERMAL PARAMETERS
      XU=BRICK(I,8)
      XF=BRICK(I,7)
      HET=BRICK(I,11)
C--- TEST WHETHER ELEMENT IS INITIALLY FROZEN OR UNFROZEN
      IF(HET.LT.0.0) GO TO 206
C--- ELEMENT IS INITIALLY THAWED
      XKX=BRICK(I,6)
      XKY=XKX
      DENSTY=BRICK(I,10)
      THER=DENSTY*XU
      NTEST=1
      GO TO 207
C--- ELEMENT IS INITIALLY FROZEN
  206 XKX=BRICK(I,5)
      XKY=XKX
      HET=-HET
      DENSTY=BRICK(I,9)
      THER=XF*DENSTY
      NTEST=-1
  207 CONTINUE
      IF(KODE1.EQ.1) I=KB
      DELX=BRICK(I,3)
      DELY=BRICK(I,4)
C---
C--- DETERMINE "BRICK" COORDINATES
C---
      CORD(1,1)=BRICK(I,1)
      CORD(1,2)=BRICK(I,2)
      CORD(2,1)=CORD(1,1)
      CORD(2,2)=CORD(1,2)-DELY
      CORD(3,1)=CORD(1,1)+DELX
      CORD(3,2)=CORD(2,2)
      CORD(4,1)=CORD(3,1)
      CORD(4,2)=CORD(1,2)
      IF(KODE1.NE.1) GO TO 208
      IF(JX-2) 210,223,223
  208 CONTINUE
C---
```

```
C--- PROCESS SUB 1
C---
  210 CONTINUE
      X(1)=CORD(1,1)
      Y(1)=CORD(1,2)
      X(2)=CORD(2,1)
      Y(2)=CORD(2,2)
      X(3)=CORD(4,1)
      Y(3)=CORD(4,2)
      DO 221 LK=1,6
      LKK=NS(1,LK)
  221 NOD(LK)=IBRICK(I,LKK)
      CALL SETUP2(XKX,XKY,THER,X,Y,NOD,NUMBER,KODE1,DENSTY,HET,XU,XF)
      IF(KODE1.EQ.1) GO TO 810
      IF(KODE1.EQ.2) GO TO 223
      NBS=0
      DO 222 LK=1,6
      LKK=NS(1,LK)
      NBR=IBRICK(I,LKK)
      IF(ACCUM(NBR).GT.0.0) NBS=NBS+1
  222 CONTINUE
      NBR=NUMBER-1
      NTHAWD(NBR)=NBS*NTEST
C---
C--- PROCESS SUB 2
C---
  223 CONTINUE
      X(1)=CORD(2,1)
      Y(1)=CORD(2,2)
      X(2)=CORD(3,1)
      Y(2)=CORD(3,2)
      X(3)=CORD(4,1)
      Y(3)=CORD(4,2)
      DO 225 LK=1,6
      LKK=NS(2,LK)
  225 NOD(LK)=IBRICK(I,LKK)
      CALL SETUP2(XKX,XKY,THER,X,Y,NOD,NUMBER,KODE1,DENSTY,HET,XU,XF)
      IF(KODE1.EQ.1) GO TO 810
      IF(KODE1.EQ.2) GO TO 228
      NBS=0
      DO 226 LK=1,6
      LKK=NS(2,LK)
      NBR=IBRICK(I,LKK)
      IF(ACCUM(NBR).GT.0.0) NBS=NBS+1
  226 CONTINUE
      NTHAWD(NUMBER)=NBS*NTEST
  228 CONTINUE
      IF(I.GE.NBRICK) GO TO 230
      GO TO 205
  230 CONTINUE
C---
C---
C--- PROCESS TETRAHEDRON/TRIANGULAR ELEMENTS
C---
C--- PROCESS TETRAHEDRON ELEMENTS
C--- MAIN PROGRAM PREPARES PARAMETERS FOR EACH TETRAHEDRON ELEMENT TO BE
```

```
C--- TRANSFERRED TO SUBROUTINE SETUP.
C---
      IF(NTETRA.EQ.0) GO TO 280
      I=0
      IF(KODE.EQ.2) GO TO 255
  231 I=I+1
      NUMBER=I
      XKX=TETRA(I,13)
      XKY=TETRA(I,14)
      XKZ=TETRA(I,15)
      THER=TETRA(I,16)
      IF(KODE1.EQ.2) GO TO 235
      HET=TETRA(I,19)
      XU=TETRA(I,16)
      XF=TETRA(I,15)
C--- TEST WHETHER ELEMENT IS INITIALLY FROZEN OR UNFROZEN
      IF(HET.LT.0.0) GO TO 233
      XKX=TETRA(I,14)
      XKY=XKX
      XKZ=XKX
      DENSTY=TETRA(I,18)
      THER=DENSTY*XU
      GO TO 235
  233 XKX=TETRA(I,13)
      XKY=XKX
      XKZ=XKX
      DENSTY=TETRA(I,17)
      HET=-HET
      THER=DENSTY*XF
  235 CONTINUE
      IF(KODE1.EQ.1) I=KKK
      X(1)= TETRA(I,1)
      Y(1)=TETRA(I,2)
      Z(1)=TETRA(I,3)
      X(2)= TETRA (I,4)
      Y(2)= TETRA (I,5)
      Z(2)= TETRA(I,6)
      X(3)= TETRA (I,7)
      Y(3)= TETRA (I,8)
      Z(3)=TETRA(I,9)
      X(4)= TETRA (I,10)
      Y(4)=TETRA(I,11)
      Z(4)= TETRA (I,12)
      DO 240 J=1,10
  240 NOD(J)=ITETRA(I,J)
      V= -1.0
      CALL SETUP(V,XKX,XKY,XKZ,THER,X,Y,Z,NOD,NUMBER,KODE1,DENSTY,HET,XU
     1,XF)
      IF(KODE1.EQ.1) GO TO 810
      IF(KODE1.EQ.2) GO TO 249
      NBS=0
      DO 242 LK=1,10
      NBR=ITETRA(I,LK)
      IF(ACCUM(NBR).GT.0.0) NBS=NBS+1
  242 CONTINUE
      NTHAWD(NUMBER)=NBS
```

```
  249 CONTINUE
      IF(I.GE.NTETRA) GO TO 280
      GO TO 231
  255 I=I+1
C---
C---
C--- PROCESS TWO-DIMENSIONAL TRIANGULAR ELEMENTS
C--- MAIN PROGRAM PREPARES PARAMETERS FOR EACH TRIANGLE ELEMENT TO BE
C--- TRANSFERRED TO SUBROUTINE SETUP2
C---
C---
      NUMBER=I
      XKX=TETRA(I,7)
      XKY=TETRA(I,8)
      THER=TETRA(I,9)
      IF(KODE1.EQ.2) GO TO 257
      HET=TETRA(I,13)
      XU=TETRA(I,10)
      XF=TETRA(I,9)
C--- TEST WHETHER ELEMENT IS INITIALLY FROZEN OR UNFROZEN
      IF(HET.LT.0.0) GO TO 256
      XKX=TETRA(I,8)
      XKY=XKX
      DENSTY=TETRA(I,12)
      THER=DENSTY*XU
      GO TO 257
  256 XKX=TETRA(I,7)
      XKY=XKX
      HET=-HET
      DENSTY=TETRA(I,11)
      THER=DENSTY*XF
  257 CONTINUE
      IF(KODE1.EQ.1) I=KKK
      X(1)=TETRA(I,1)
      Y(1)=TETRA(I,2)
      X(2)=TETRA(I,3)
      Y(2)=TETRA(I,4)
      X(3)=TETRA(I,5)
      Y(3)=TETRA(I,6)
      DO 260 J=1,6
  260 NOD(J)=ITETRA(I,J)
      CALL SETUP2(XKX,XKY,THER,X,Y,NOD,NUMBER,KODE1,DENSTY,HET,XU,XF)
      IF(KODE1.EQ.1) GO TO 810
      IF(KODE1.EQ.2) GO TO 277
      NBS=0
      DO 262 LK=1,6
      NBR=ITETRA(I,LK)
      IF(ACCUM(NBR).GT.0.0) NBS=NBS+1
  262 CONTINUE
      NTHAMD(NUMBER)=NBS
  277 CONTINUE
      IF(I.GE.NTETRA) GO TO 280
      GO TO 255
  280 CONTINUE
C---
C---
```

```
C--- PREPARE "H" AND "Z" MATRICES FOR TIME ADVANCEMENT SCHEME
C--- MODIFYING THE "GK" AND "GC" MATRICES RESPECTIVELY.
C---
C--- THETA= TIME STEP INCREMENT
C---
      DO 700  I=1, NNODES
      DO 700   J=1, NBAND
      GC(I,J)=2.0*GC(I,J)/THETA
      GK(I,J)= GK(I,J) + GC(I,J)
      GC(I,J)= 2.0* GC(I,J)- GK(I,J)
  700 CONTINUE
C*
C**
C***
C**********SEGMENT-FOUR-----INSERT BOUNDARY CONDITIONS**********
C***
C**
C*
C---
C---
C---SET TT(I) EQUAL TO ZERO
C---
C---
      DO 800 I=1,NNODES
  800 TT(I)=0.0
C---
C--- MODIFY GLOBAL MATRICES
C---
C---
C---   ACCOMODATE UPPER TRIANGLE TERMS
C---
  810 CONTINUE
      IF(KODE2.EQ.1) GO TO 990
      IF(KODE1.EQ.1) GO TO 995
  811 CONTINUE
C--- IF IN TIME ADVANCEMENT ROUTINE, REAPPLY BOUNDARY CONDITIONS AFTER
C--- EVERY ELEMENT PHASE CHANGE
      DO 850 I=1, NUMBC
      K= NBC(I)
      XF= BC(I)
      GK(K,1)=1.0
      GC(K,1)=1.0
      BEGIN(K)=XF
      IF (K.EQ.1) GO TO 830
      KA= K+1
      KP=K
      IF(K.GT.NBAND) KP=NBAND
      DO 820  J=2, KP
      KK=KA-J
      TT(KK)=TT(KK)-GK(KK,J)*XF+GC(KK,J)*XF
      GK(KK,J)=0.0
      GC (KK,J) = 0.0
  820 CONTINUE
C---
C---   ACCOMODATE LOWER TRIANGLE TERMS USING SYMMETRY
C---    OF ELEMENT MATRIX
```

```
C---
  830 IF (K.EQ.NNODES) GO TO 850
      KA=K-1
      KP= NNODES-KA
      IF(KP.GT.NBAND) KP= NBAND
      DO 840  J=2, KP
      KK= KA+J
      TT(KK)=TT(KK)-GK(K,J)*XF+GC(K,J)*XF
      GK(K,J)= 0.0
      GC(K,J) = 0.0
  840 CONTINUE
  850 CONTINUE
      DO 860 I=1,NUMBC
      K=NBC(I)
  860 TT(K)=0.0
      IF(KODE3.EQ.1) GO TO 1005
C*
C**
C***
C**********SEGMENT-FIVE-----TIME ADVANCEMENT ROUTINE**********
C***
C**
C*
C---
C---- CRANK-NICOLSEN TIME ADVANCEMENT
C---
C---
      IF(KODE1.EQ.2) GO TO 871
C--- FLAG KODE1 THAT PHASE CHANGE APPROXIMATION IS EMPLOYED, AND THAT
C--- PROGRAM IS NOW IN THE TIME ADVANCEMENT ROUTINE
      KODE1=1
C--- UTILIZE "DENSITY" STORAGE SPACE TO STORE "THETA" IN THE TIME ADVANCEMENT
C--- ROUTINE
      DENSITY=THETA
C---
C--- DERIVE "ACCUM(I)",CU(I) AND CF(I) VECTORS
C---
C--- USE "HEAT(I)" AS A FLAG TO INDICATE WHAT PHASE NODE EXISTS IN.
C--- A POSITIVE "HEAT(I)" IMPLIES NODE EXISTS IN UNFROZEN STATE.
C--- A NEGATIVE "HEAT(I)" IMPLIES THE NODE EXISTS IN FROZEN STATE.
      DO 870 I=1,NNODES
      CU(I)=CU(I)/WEIGHT(I)
      CF(I)=CF(I)/WEIGHT(I)
      HEAT(I)=-HEAT(I)
      IF(ACCUM(I).EQ.0.0) GO TO 870
      HEAT(I)=-HEAT(I)
      ACCUM(I)=HEAT(I)
  870 CONTINUE
  871 CONTINUE
      TIME=0.0
      COUNTI=0.0
  890 KODE3=0
      DO 895 I=1,NNODES
      DO 895 J=1,NBAND
  895 SAVE(I,J)=GK(I,J)
      CALL PRESOL(NNODES , NBAND)
```

```
  900 CALL COMB(BEGIN,NNODES,NBAND)
      DO 910 I=1,NNODES
  910 ZZ(I)=ZZ(I)+TT(I)
      CALL FINSUL(NNODES, NBAND)
      TIME= TIME+ THETA
      COUNTI= COUNTI+1.0
      IF(COUNTI.LT.3.0) GO TO 1001
      WRITE(3,10)TIME,COUNTI
      WRITE(3,15)
      WRITE(3,16)(I,ZZ(I),I=1,NNODES)
      WRITE(3,20)
      COUNTI=0.0
      IF(KODE1.EQ.2) GO TO 1005
      WRITE(3,21)
      WRITE(3,16)(I,ACCUM(I),I=1,NNODES)
      WRITE(3,21)
C---
C--- PHASE CHANGE SIMULATION
C---
C---
C--- HEAT(I) = HEAT OF FUSION FOR NODE "I"
C--- ACCUM(I) = LATENT HEAT ACCUMULATOR FOR NODE "I"
C---
C--- TEST EACH NODE FOR POSSIBLE PHASE CHANGE
      L=0
  913 L=L+1
      XF=ZZ(L)-FPD
C--- DETERMINE WHAT PHASE STATE NODE "L" EXISTS
      IF(HEAT(L))920,920,915
C--- NODE EXISTS IN UNFROZEN STATE
  915 IF(XF)916,916,1000
C--- CALCULATE QUANTITY OF LATENT HEAT INVOLVED
  916 Q=CU(L)*XF*WEIGHT(L)
      ACCUM(L)=ACCUM(L)+Q
C--- TEST IF REQUISITE AMOUNT OF LATENT HEAT OF NODE "L" IS EXHAUSTED
      IF(ACCUM(L))917,917,919
C--- FREEZING OF NODE OCCURS
  917 QQ=ACCUM(L)
      WRITE(3,2500)L
      ZZ(L)=QQ/CF(L)/WEIGHT(L)+FPD
      ACCUM(L)=0.0
      HEAT(L)=-HEAT(L)
      NDELTA=-1
      GO TO 930
C--- NODE "L" IS SET AT FREEZING POINT DEPRESSION (FPD)
  919 ZZ(L)=FPD
      GO TO 1000
C--- NODE EXISTS IN FROZEN STATE
  920 IF(XF)1000,921,921
C--- CALCULATE LATENT HEAT TO BE ADDED TO ACCUMULATOR FOR NODE "L"
  921 Q=CF(L)*XF*WEIGHT(L)
      ACCUM(L)=ACCUM(L)+Q
C--- TEST IF ENOUGH HEAT HAS BEEN ADDED TO CAUSE NODE "L" TO THAW
      XF=-HEAT(L)
      IF(ACCUM(L).GE.XF) GO TO 922
C--- NODE "L" IS SET AT FREEZING POINT DEPRESSION (FPD)
```

```
      ZZ(L)=FPD
      GO TO 1000
C---- THAWING OF NODE OCCURS
  922 WRITE(3,2501)L
      QQ=ACCUM(L)+HEAT(L)
      ZZ(L)=QQ/CU(L)/WEIGHT(L)+FPD
      HEAT(L)=-HEAT(L)
      ACCUM(L)=HEAT(L)
      NDELTA=1
  930 CONTINUE
C---- PHASE CHANGE HAS OCCURED AT NODE "L".
C---- DETERMINE ALL ELEMENTS ASSOCIATED TO NODE "L".
      KKKK=MATRIX(L,1)-1
      KKX=1
  932 KKX=KKX+1
      KKK=MATRIX(L,KKX)
C---- KODE2=1 FLAGS ROUTINE TO RETURN TO BRICK-SUBELEMENT TESTING SEGMENT
C---- KODE2=0 IMPLIES THAT ALL BRICK-SUBELEMENTS HAVE BEEN TESTED
      KODE2=0
C---- TEST IF ELEMENT KKK IS A BRICK TYPE ELEMENT.
      IF(KKK-NTETRA)934,934,970
C---- ELEMENT KKK IS A TETRA TYPE ELEMENT
C---- TEST IF NEW PHASE OF NODE "L" MATCHES PHASE OF ELEMENT. IF THE PHASES ARE
C---- SIMILAR, DO NOT CHANGE PHASE OF THE ELEMENT.
  934 XF=TETRA(KKK,19)
      IF(KODE.EQ.2) XF=TETRA(KKK,13)
      NTHAWD(KKK)=NTHAWD(KKK)+NDELTA
C---- A NEGATIVE LATENT HEAT (XF) IMPLIES ELEMENT IS FROZEN
      IF(ACCUM(L).GT.0.0) GO TO 935
C---- NODE "L" IS NEWLY FROZEN
      IF(XF.LT.0.0) GO TO 995
C---- TEST IF ELEMENT "KKK" FREEZES
      IF(NTHAWD(KKK).GT.0) GO TO 995
C---- ELEMENT "KKK" FREEZES
      TETRA(KKK,19)=-XF
      IF(KODE.EQ.2) TETRA(KKK,13)=-XF
      GO TO 950
C---- NODE "L" IS NEWLY THAWED
  935 IF(XF.GT.0.0) GO TO 995
C---- TEST IF ELEMENT "KKK" THAWS
      NTEST=10
      IF(KODE.EQ.2) NTEST=6
      IF(NTHAWD(KKK).LT.NTEST) GO TO 995
C---- ELEMENT "KKK" THAWS
      TETRA(KKK,19)=-XF
      IF(KODE.EQ.2) TETRA(KKK,13)=-XF
  950 IF(KODE.EQ.2) GO TO 955
C---- DETERMINE 3-D THERMAL PARAMETER CHANGES
      XKX=TETRA(KKK,14)-TETRA(KKK,13)
      THER=TETRA(KKK,16)*TETRA(KKK,18)-TETRA(KKK,15)*TETRA(KKK,17)
      GO TO 960
C---- DETERMINE 2-D THERMAL PARAMETER CHANGES
  955 XKX=TETRA(KKK,10)-TETRA(KKK,9)
      THER=TETRA(KKK,10)*TETRA(KKK,12)-TETRA(KKK,9)*TETRA(KKK,11)
  960 CONTINUE
      IF(KODE3.EQ.1) GO TO 968
```

```
      DO 965 I=1,NNODES
      DO 965 J=1,MBAND1
  965 GK(I,J)=SAVE(I,J)
      KODE3=1
  968 CONTINUE
      WRITE(3,2502)XKK
      IF(ACCUM(L).LE.0.0) XKX=-XKX
      IF(ACCUM(L).LE.0.0) THERE=-THER
      XKY=XKX
      XKZ=XKX
C--- TRANSFER, AND MODIFY GLOBAL MATRICES.
C--- (ROUTINE WILL RETURN TO STEP 995)
      IF(KODE-2)257,257,255
C--- ELEMENT XKK IS A BRICK TYPE ELEMENT
  970 KD=2
      KE=6
      JX=1
      KODE2=1
      IF(KODE.EQ.3) GO TO 971
C--- 2-D "BRICK" ELEMENTS ARE NUMBERED AS 12,14,16,..., TO ACCOUNT FOR
C--- INTER-"BRICK" TRIANGLE ELEMENTS.
      KB=(KKK-NTETRA)/2
C--- "KB" = CORRESPONDING ROW IN "IBRICK" AND "BRICK" DATA BLOCKS
      GO TO 972
C--- 3-D "BRICK" ELEMENTS ARE NUMBERED AS 15,20,25,..., WHERE
C--- SUBELEMENT NUMBER FIVE IS THE INTERIOR TETRAHEDRON
  971 KB=(KKK-NTETRA)/5
      KD=5
      KE=10
  972 CONTINUE
C--- TEST EACH SUBELEMENT WHETHER IT IS ASSOCIATED TO NODE "L"
      K=1
  973 KA=NS(JX,K)
      KC=IBRICK(KD,KA)
      IF(L.EQ.KC) GO TO 974
      K=K+1
      IF(K.LE.KE) GO TO 973
      GO TO 990
C--- DETERMINE SUBELEMENT GLOBAL ELEMENT NUMBER "JJ"
  974 JJ=XKK-KD+JX
C--- UPDATE NTHAND(JJ)
      NTEST=1
      IF(NTHAND(JJ).LE.0) NTEST=-1
      NTHAND(JJ)=NTHAND(JJ)+NTESTANDELTA
C--- TEST IF SUBELEMENT "JJ" IS OF SAME PHASE AS NODE "L"
      IF(NTEST*NDELTA.EQ.1) GO TO 990
C--- TEST IF SUBELEMENT "JJ" CHANGES PHASE
      IF(NTHAND(JJ).EQ.0) GO TO 975
      IF(NTHAND(JJ).EQ.-KE) GO TO 975
      GO TO 990
C--- SUBELEMENT CHANGES PHASE
  975 IF(KODE.EQ.3) GO TO 976
C--- ELEMENT IS A 2-D BRICK ELEMENT.
C--- DERIVE THERMAL PARAMETER CHANGES
      XKX=BRICK(KB,6)-BRICK(KB,5)
      THERE=BRICK(KB,8)*BRICK(KB,10)-BRICK(KB,7)*BRICK(KB,9)
```

```
      GO TO 977
C--- ELEMENT IS A 3-D BRICK ELEMENT.
C--- DERIVE THERMAL PARAMETER CHANGES
  976 XKX=BRICK(KB,8)-BRICK(KB,7)
      THER=BRICK(KB,10)*BRICK(KB,12)-BRICK(KB,9)*BRICK(KB,11)
  977 CONTINUE
      IF(ACCUM(L).LE.0.0) XKX=-XKX
      IF(ACCUM(L).LE.0.0) THER=-THER
      XKY=XKX
      XKZ=XKX
      WRITE(3,2502)JJ
      IF(KODE3.EQ.1) GO TO 985
      DO 980 I=1,NNODES
      DO 980 J=1,NBAND
  980 GK(I,J)=SAVE(I,J)
      KODE3=1
  985 CONTINUE
C--- UPDATE ELEMENT PHASE INDICATOR
      NTHAND(JJ)=-NTHAND(JJ)
      IF(KODE.EQ.2) GO TO 207
      GO TO 145
  990 IF(JX.GE.KB) GO TO 995
      JX=JX+1
      GO TO 972
  995 KODE2=0
      IF(KKX.GE.KKKK) GO TO 1000
      GO TO 932
 1000 IF(L.GE.NNODES) GO TO 1001
      GO TO 913
 1001 CONTINUE
C--- RESET BOUNDARY CONDITIONS
      IF(KODE3.EQ.0) GO TO 1002
      GO TO 811
 1002 DO 1003 I=1,NBC
      K=NBC(I)
 1003 ZZ(K)=BC(I)
 1005 CONTINUE
      DO 1995 I=1,NNODES
 1995 BEGIN(I)=ZZ(I)
      IF(TIME.GE.DAYS) GO TO 2000
      IF(KODE3.EQ.1) GO TO 890
      GO TO 900
 2000 CONTINUE
 2500 FORMAT(3X,'NODE',I3,1X,'FREEZES')
 2501 FORMAT(3X,'NODE',I3,1X,'THAWS')
 2502 FORMAT(3X,'ELEMENT',I3,1X,'CHANGES PHASE')
 2503 FORMAT(10X,'TRANSIENT HEAT CONDUCTION MODEL WITH ISOTHERMAL PHASE
     1CHANGE APPROXIMATION')
 2504 FORMAT(10X,'TRANSIENT HEAT CONDUCTION MODEL')
      STOP
      END
```

```
      SUBROUTINE SETUP(V,XKX,XKY,XKZ,THER,X,Y,Z,NOD,NUMBER,KODE1,DENSITY,
     1HET,XC,XX)
C---
C---  THIS SUBROUTINE COMPUTES THE ELEMENT CONDUCTION AND CAPACITANCE
C---  MATRICES FOR THE 3-D PROBLEM. THE RESULTS ARE PLACED INTO THE
C---  GLOBAL MATRIX SYSTEM.
C---
C---
      COMMON/BLK 1/GK(100,40)
      COMMON/BLK 2/GC(100,40)
      COMMON/BLK 5/WEIGHT(100)
      COMMON/BLK 6/MATRIX(100,5)
      COMMON/BLK 7/ HEAT(100)
      COMMON/BLK 8/ CO(100)
      COMMON/ BLK 9/ CF(100)
      DIMENSION XK(10,10),XC(10,10),XTEMP(10,10),X(4),Y(4),Z(4),INDEX(10
     1),INDEM(10),NOD(10)
C---
C---
C---DERIVE COFACTORS OF 4X4 DETERMINANT RELATING GLOBAL COORDINATES
C---TO LOCAL COORDINATES.
C---
C---
      AA1=-Y(3)*Z(4)+Z(3)*Y(4)+Y(2)*Z(4)-Z(2)*Y(4)-Y(2)*Z(3)+Z(2)*Y(3)
      AA2=Y(3)*Z(4)-Y(1)*Z(4)+Z(1)*Y(4)+Y(1)*Z(3)-Z(1)*Y(3)-Y(4)*Z(3)
      AA3=-Y(2)*Z(4)+Z(2)*Y(4)+Y(1)*Z(4)-Z(1)*Y(4)-Y(1)*Z(2)+Z(1)*Y(2)
      AA4=Y(2)*Z(3)-Z(2)*Y(3)-Y(1)*Z(3)+Z(1)*Y(3)+Y(1)*Z(2)-Z(1)*Y(2)
      BB1=X(3)*Z(4)-X(4)*Z(3)-X(2)*Z(4)+X(4)*Z(2)+X(2)*Z(3)-X(3)*Z(2)
      BB2=-X(3)*Z(4)+X(4)*Z(3)+X(1)*Z(4)-X(4)*Z(1)-X(1)*Z(3)+X(3)*Z(1)
      BB3=X(2)*Z(4)-X(4)*Z(2)-X(1)*Z(4)+X(4)*Z(1)+X(1)*Z(2)-X(2)*Z(1)
      BB4=-X(2)*Z(3)+X(3)*Z(2)+X(1)*Z(3)-X(3)*Z(1)-X(1)*Z(2)+X(2)*Z(1)
      C1=-X(3)*Y(4)+X(4)*Y(3)+X(2)*Y(4)-X(4)*Y(2)-X(2)*Y(3)+X(3)*Y(2)
      C2=X(3)*Y(4)-X(4)*Y(3)-X(1)*Y(4)+X(4)*Y(1)+X(1)*Y(3)-X(3)*Y(1)
      C3=-X(2)*Y(4)+X(4)*Y(2)+X(1)*Y(4)-X(4)*Y(1)-X(1)*Y(2)+X(2)*Y(1)
      C4=X(2)*Y(3)-X(3)*Y(2)-X(1)*Y(3)+X(3)*Y(1)+X(1)*Y(2)-X(2)*Y(1)
C---
C---CALCULATE TETRAHEDRON ELEMENT VOLUME (IF NECESSARY)
C---
      IF(V.GT.0.0) GO TO 399
      V=X(1)*AA1+X(2)*AA2+X(3)*AA3+X(4)*AA4
      V=V/6.0
  399 CONTINUE
      IF(KODE1.NE.0) GO TO 401
C--- SETUP MATRIX(I,J) POINTER ARRAY AND NODAL PARAMETER VECTORS.
      VV=V*DENSITY/10.0
      DO 400 J=1,10
      K=NOD(J)
      HEAT(K)=HEAT(K)+VV*HET
      WEIGHT(K)=WEIGHT(K)+VV
      CO(K)=CO(K)+XC*VV
      CF(K)=CF(K)+XX*VV
      L=MATRIX(K,1)
      IF(L.EQ.2) GO TO 396
      LA=L-1
      IF(NUMBER.EQ.MATRIX(K,LA)) GO TO 400
  396 CONTINUE
```

```
      MATRIX(K,L)=(THEN
      MATRIX(K,1)=L+1
  400 CONTINUE
  401 CONTINUE
C---
C---
C---
C---DERIVE TETRAHEDRON ELEMENT CONDUCTION MATRIX "XK"
C---
      XF=180.0*V
      S1=(XKX*AA1*AA1+XKY*BB1*BB1+XKZ*C1*C1)/XF
      S2=(XKX*AA2*AA2+XKY*BB2*BB2+XKZ*C2*C2)/XF
      S3=(XKX*AA3*AA3+XKY*BB3*BB3+XKZ*C3*C3)/XF
      S4=(XKX*AA4*AA4+XKY*BB4*BB4+XKZ*C4*C4)/XF
      XK(1,4)=-(XKX*AA1*AA4+XKY*BB1*BB4+XKZ*C1*C4)/XF
      XK(1,3)=-(XKX*AA1*AA3+XKY*BB1*BB3+XKZ*C1*C3)/XF
      XK(1,2)=-(XKX*AA1*AA2+XKY*BB1*BB2+XKZ*C1*C2)/XF
      XK(1,5)=-3.0*XK(1,2)-S1
      XK(1,6)=-3.0*XK(1,3)-S1
      XK(1,7)=-3.0*XK(1,4)-S1
      XK(1,1)=3.0*S1
      XK(1,8)=XK(1,2)+XK(1,5)
      XK(1,9)=XK(1,3)+XK(1,4)
      XK(1,10)=XK(1,2)+XK(1,4)
      XK(2,2)=3.0*S2
      XK(2,3)=-(XKX*AA2*AA3+XKY*BB2*BB3+XKZ*C2*C3)/XF
      XK(2,4)=-(XKX*AA2*AA4+XKY*BB2*BB4+XKZ*C2*C4)/XF
      XK(2,5)=-3.0*XK(1,2)-S2
      XK(2,6)=XK(1,2)+XK(2,5)
      XK(2,7)=XK(1,2)+XK(2,4)
      XK(2,8)=-3.0*XK(2,3)-S2
      XK(2,9)=XK(2,5)+XK(2,4)
      XK(2,10)=-3.0*XK(2,4)-S2
      XK(3,3)=3.0*S3
      XK(3,4)=-(XKX*AA4*AA3+XKY*BB3*BB4+XKZ*C3*C4)/XF
      XK(3,5)=XK(1,5)+XK(2,5)
      XK(3,6)=-3.0*XK(1,3)-S3
      XK(3,7)=XK(1,3)+XK(3,4)
      XK(3,8)=-3.0*XK(2,3)-S3
      XK(3,9)=-3.0*XK(3,4)-S3
      XK(3,10)=XK(2,3)+XK(3,4)
      XK(4,4)=3.0*S4
      XK(4,5)=XK(1,4)+XK(2,4)
      XK(4,6)=XK(1,4)+XK(3,4)
      XK(4,7)=-3.0*XK(1,4)-S4
      XK(4,8)=XK(2,4)+XK(3,4)
      XK(4,9)=-3.0*XK(3,4)-S4
      XK(4,10)=-3.0*XK(2,4)-S4
      XK(5,5)=8.0*(S1+S2-XK(1,2))
      XK(5,6)=4.0*(S1-2.0*XK(2,3)-XK(1,8))
      XK(5,7)=4.0*(S1-2.0*XK(2,4)-XK(1,10))
      XK(5,8)=4.0*(S2-2.0*XK(1,3)-XK(2,8))
      XK(5,9)=-4.0*(XK(1,9)+XK(2,9))
      XK(5,10)=4.0*(S2-2.0*XK(1,4)-XK(2,7))
      XK(6,6)=8.0*(S1+S3-XK(1,3))
      XK(6,7)=4.0*(S1-2.0*XK(3,4)-XK(1,9))
```

```
      XK(5,5)=4.0*(S3-2.0*XK(1,2)-XK(3,5))
      XK(5,9)=4.0*(S3-2.0*XK(1,4)-XK(3,7))
      XK(6,10)=-4.0*(XK(1,10)+XK(3,10))
      XK(7,7)=8.0*(S1-XK(1,4)+S4)
      XK(7,8)=-4.0*(XK(1,8)+XK(4,8))
      XK(7,9)=4.0*(S4-2.0*XK(1,3)-XK(4,6))
      XK(7,10)=4.0*(S4-2.0*XK(1,2)-XK(4,5))
      XK(8,8)=8.0*(S2+S3-XK(2,5))
      XK(8,9)=4.0*(S3-2.0*XK(2,4)-XK(3,10))
      XK(8,10)=4.0*(S2-2.0*XK(3,4)-XK(2,9))
      XK(9,9)=8.0*(S3+S4-XK(3,4))
      XK(9,10)=4.0*(S4-2.0*XK(2,3)-XK(4,8))
      XK(10,10)=8.0*(S2+S4-XK(2,4))
C---
C---
C---FORM CAPACITANCE MATRIX "XC" FOR TETRAHEDRON ELEMENT
C---
C---
C---COMPUTE "XC" MATRIX CONSTANTS
C---
C---
      A1=1.0/70.0
      A2= 1.0/ 420.0
      A3= -1.0/105.0
      A4= -1.0/70.0
      B3= 8.0/105.0
      B4= 4.0/105.0
      B5= 2.0/105.0
      XF=THER*V
      AL1=A1*XF
      AL2=A2*XF
      AL3=A3*XF
      AL4=A4*XF
      BE3=B3*XF
      BE4=B4*XF
      BE5=B5*XF
C---
C---
C---DERIVE TETRAHEDRON ELEMENT CAPACITANCE MATRIX "XC"
C---
C---
      XC(1,1)=AL1
      XC(1,2)=AL2
      XC(1,3)=AL2
      XC(1,4)=AL2
      XC(1,5)=AL3
      XC(1,6)=AL3
      XC(1,7)=AL3
      XC(1,8)=AL4
      XC(1,9)=AL4
      XC(1,10)=AL4
      XC(2,2)=AL1
      XC(2,3)=AL2
      XC(2,4)=AL2
      XC(2,5)=AL3
      XC(2,6)=AL4
```

```
      XC(2,7)=AL4
      XC(2,8)=AL3
      XC(2,9)=AL4
      XC(2,10)=AL3
      XC(3,3)=AL1
      XC(3,4)=AL2
      XC(3,5)=AL4
      XC(3,6)=AL3
      XC(3,7)=AL4
      XC(3,8)=AL3
      XC(3,9)=AL3
      XC(3,10)=AL4
      XC(4,4)=AL1
      XC(4,5)=AL4
      XC(4,6)=AL4
      XC(4,7)=AL3
      XC(4,8)=AL4
      XC(4,9)=AL3
      XC(4,10)=AL3
      XC(5,5)=BE3
      XC(5,6)=BE4
      XC(5,7)=BE4
      XC(5,8)=BE4
      XC(5,9)=BE5
      XC(5,10)=BE4
      XC(6,6)=BE3
      XC(6,7)=BE4
      XC(6,8)=BE4
      XC(6,9)=BE4
      XC(6,10)=BE5
      XC(7,7)=BE3
      XC(7,8)=BE5
      XC(7,9)=BE4
      XC(7,10)=BE4
      XC(8,8)=BE3
      XC(8,9)=BE4
      XC(8,10)=BE4
      XC(9,9)=BE3
      XC(9,10)=BE4
      XC(10,10)=BE3
C---
C---
C---SUBSTITUTE ELEMENT MATRIX INTO GLOBAL MATRIX
C---
C---
C---
C---FILL IN LOWER TRIANGLE OF ELEMENT "XK" MATRIX
C---
C---
      DO 420 I=1,9
      J=I+1
      DO 410 K=1,I
      XC(J,K)=XC(K,J)
  410 XK(J,K)=XK(K,J)
  420 CONTINUE
      IF(KODE1.NE.1) GO TO 429
```

```
      THETA=DENSTY
      DO 425 I=1,10
      DO 425 J=1,10
      XC(I,J)=2.0*XC(I,J)/THETA
      XK(I,J)=XK(I,J)+XC(I,J)
      XC(I,J)=2.0*XC(I,J)-XK(I,J)
  425 CONTINUE
C---
C---
C---ARRANGE NODE SEQUENCE INTO NUMERICAL ORDER, AND PUT INTO "INDEM".
C---ARRANGE ROW/COLUMN POINTER ARRAY "INDEX" INTO  ORDER ANALOGOUS TO
C---"INDEM" SEQUENCE.
C---
C---
  429 CONTINUE
      DO 430 I=1,10
      INDEX(I)=I
  430 INDEM(I)=NOD(I)
      DO 440 I=1,9
      K=I+1
      DO 440 J=K,10
      IF(INDEM(J).GT.INDEM(I)) GO TO 440
      NTEMM=INDEX(J)
      NTEMP=INDEM(J)
      INDEX(J)=INDEX(I)
      INDEM(J)=INDEM(I)
      INDEX(I)=NTEMM
      INDEM(I)=NTEMP
  440 CONTINUE
C---
C---
C---REARRANGE ROWS AND COLUMNS OF ELEMENT MATRIX INTO NUMERICAL ORDER
C---
C---
C---REARRANGE ROWS OF "XK" AND PUT INTO "XTEMP":
C---
      DO 450 I=1,10
      DO 450 J=1,10
      K=INDEX(I)
  450 XTEMP(I,J)=XK(K,J)
C---
C---REARRANGE COLUMNS OF "XTEMP" AND PUT INTO "XK":
C---
      DO 460 I=1,10
      DO 460 J=1,10
      K=INDEX(I)
  460 XK(J,I)=XTEMP(J,K)
C---
C---
C---PUT "XK" ELEMENT MATRIX INTO BANDED FORM GLOBAL MATRIX "GK":
C---
C---
      DO 470 I=1,10
      IROW=INDEM(I)
      II=IROW-1
      DO 470 J=I,10
```

```
      ICOL=INDEX(J)-11
  470 GK(IROW,ICOL)=GK(IROW,ICOL)+XK(I,J)
C---
C---
C---SUBSTITUTE ELEMENT MATRIX "XC" INTO GLOBAL MATRIX "GC".
C---
C---
C---
C---REARRANGE ROWS AND COLUMNS OF ELEMENT "XC" MATRIX INTO NUMERICAL
C---ORDER, IN THE SAME METHOD THAT THE ELEMENT MATRIX "XK" WAS ORDERED.
C---
C---
C---REARRANGE ROWS OF "XC" AND PUT INTO "XTEMP":
C---
      DO 500 I=1,10
      DO 500 J=1,10
      K=INDEX(I)
  500 XTEMP(I,J)=XC(K,J)
C---
C---
C---REARRANGE COLUMNS OF "XTEMP" AND PUT INTO "XC":
C---
      DO 510 I=1,10
      DO 510 J=1,10
      K=INDEX(I)
  510 XC(J,I)=XTEMP(J,K)
C---
C---
C---PUT ELEMENT "XC" MATRIX INTO GLOBAL "GC" MATRIX.
C---
C---
      DO 520 I=1,10
      IROW=INDEX(I)
      II=I-N+1
      DO 520 J=I,10
      ICOL=INDEX(J)-11
  520 GC(IROW,ICOL)=GC(IROW,ICOL)+XC(I,J)
      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SETUP2(XKX,XKY,THER,X,Y,NOD,NUMBER,KODE1,DENSTY,HEI,XU,
     1XX)
C---
C--- THIS SUBROUTINE COMPUTES THE ELEMENT CONDUCTION AND CAPACITANCE
C--- MATRICES FOR THE 2-D PROBLEM. THE RESULTS ARE PLACED INTO THE
C--- GLOBAL MATRIX SYSTEM.
C---
C---
      COMMON/BLK 1/GK(100,40)
      COMMON/BLK 2/GC(100,40)
      COMMON/BLK 5/ WEIGHT(100)
      COMMON/BLK 6/ MATRIX(100,5)
      COMMON/BLK 7/ HEAT(100)
      COMMON/BLK 8/ CU(100)
      COMMON/ BLK 9/ CF(100)
      DIMENSION XK(6,6),XC(6,6),XTEMP(6,6),X(4),Y(4),INDEX(6),INDEM(6),
     1NOD(10)
C---
C---
C--- PRELIMINARY CALCULATIONS FOR ELEMENT MATRICES
C---
C---
      AA1=X(3)-X(2)
      AA2=X(1)-X(3)
      AA3=X(2)-X(1)
      BB1=Y(2)-Y(3)
      BB2=Y(3)-Y(1)
      BB3=Y(1)-Y(2)
C---
C--- CALCULATE ELEMENT AREA
C---
      V=(AA3*BB2-AA2*BB3)/2.0
      IF(KODE1.NE.0) GO TO 401
C--- SETUP MATRIX(I,J) POINTER ARRAY AND NODE PARAMETER VECTORS.
      VV=V*DENSTY/6.0
      DO 400 J=1,6
      K=NOD(J)
      HEAT(K)=HEAT(K)+VV*HEI
      WEIGHT(K)=WEIGHT(K)+VV
      CU(K)=CU(K)+XU*VV
      CF(K)=CF(K)+XX*VV
      L=MATRIX(K,1)
      IF(L.EQ.2) GO TO 398
      LA=L-1
      IF(NUMBER.EQ.MATRIX(K,LA)) GO TO 400
  398 CONTINUE
      MATRIX(K,L)=NUMBER
      MATRIX(K,1)=L+1
  400 CONTINUE
  401 CONTINUE
C---
C---
C--- CONSTRUCT ELEMENT "XC" CAPACITANCE MATRIX
C---
C---
      XF=V*THER*2.0
```

```
      AL1=XF/60.0
      AL2=-AL1/6.0
      AL3=-XF/90.0
      AL4=4.0*XF/45.0
      AL5=AL4/2.0
      XC(1,1)=AL1
      XC(1,2)=AL2
      XC(1,3)=AL2
      XC(1,4)=0.0
      XC(1,5)=AL3
      XC(1,6)=0.0
      XC(2,2)=AL1
      XC(2,3)=AL2
      XC(2,4)=0.0
      XC(2,5)=0.0
      XC(2,6)=AL3
      XC(3,3)=AL1
      XC(3,4)=AL3
      XC(3,5)=0.0
      XC(3,6)=0.0
      XC(4,4)=AL4
      XC(4,5)=AL5
      XC(4,6)=AL5
      XC(5,5)=AL4
      XC(5,6)=AL5
      XC(6,6)=AL4
C---
C---
C---  CONSTRUCT ELEMENT "XK" CONDUCTION MATRIX
C---
C---
      XF=12.0*V
      S1=(XKX*BB1*BB1+XKY*AA1*AA1)/XF
      S2=(XKX*BB2*BB2+XKY*AA2*AA2)/XF
      S3=(XKX*BB3*BB3+XKY*AA3*AA3)/XF
      XK(1,1)=3.0*S1
      XK(1,2)=-(XKX*BB1*BB2+XKY*AA1*AA2)/XF
      XK(1,3)=-(XKX*BB1*BB3+XKY*AA1*AA3)/XF
      XK(1,4)=-4.0*XK(1,2)
      XK(1,5)=0.0
      XK(1,6)=-4.0*XK(1,3)
      XK(2,2)=3.0*S2
      XK(2,3)=-(XKX*BB2*BB3+XKY*AA2*AA3)/XF
      XK(2,4)=-4.0*XK(1,2)
      XK(2,5)=-4.0*XK(2,3)
      XK(2,6)=0.0
      XK(3,3)=3.0*S3
      XK(3,4)=0.0
      XK(3,5)=-4.0*XK(2,3)
      XK(3,6)=-4.0*XK(1,3)
      XK(4,4)=8.0*(S2+S1-XK(1,2))
      XK(4,5)=4.0*(S2-2.0*XK(1,3)-XK(2,3)-XK(1,2))
      XK(4,6)=4.0*(S1-2.0*XK(2,3)-XK(1,2)-XK(1,3))
      XK(5,5)=8.0*(S3+S2-XK(2,3))
      XK(5,6)=4.0*(S3-2.0*XK(1,2)-XK(1,3)-XK(2,3))
      XK(6,6)=8.0*(S1+S3-XK(1,3))
```

```
C---
C---
C--- SUBSTITUTE ELEMENT MATRIX INTO GLOBAL MATRIX
C---
C---
C---
C--- FILL IN LOWER TRIANGLE OF ELEMENT MATRICES
C---
      DO 420 I=1,5
      J=I+1
      DO 410 K=1,I
      XK(J,K)=XK(K,J)
  410 XC(J,K)=XC(K,J)
  420 CONTINUE
      IF(KODE1.NE.1) GO TO 429
      THETA=DENSTY
      DO 425 I=1,6
      DO 425 J=1,6
      XC(I,J)=2.0*XC(I,J)/THETA
      XK(I,J)=XK(I,J)+XC(I,J)
      XC(I,J)=2.0*XC(I,J)-XK(I,J)
  425 CONTINUE
  429 CONTINUE
C---
C--- ARRANGE NODE SEQUENCE INTO NUMERICAL ORDER, AND PUT INTO "INDEM".
C--- ARRANGE ROW/COLUMN POINTER ARRAY "INDEX" INTO ORDER ANALOGOUS TO
C--- "INDEM" SEQUENCE.
C---
      DO 430 I=1,6
      INDEX(I)=I
  430 INDEM(I)=NOD(I)
      DO 440 I=1,5
      K=I+1
      DO 440 J=K,6
      IF(INDEM(J).GT.INDEM(I)) GO TO 440
      NTEMM=INDEX(J)
      NTEMP=INDEM(J)
      INDEX(J)=INDEX(I)
      INDEM(J)=INDEM(I)
      INDEX(I)=NTEMM
      INDEM(I)=NTEMP
  440 CONTINUE
C---
C---
C--- SUBSTITUTE ELEMENT "XK" MATRIX INTO GLOBAL "GK" MATRIX.
C--- REARRANGE ROWS AND COLUMNS OF ELEMENT MATRIX INTO NUMERICAL ORDER
C---
C---
C--- REARRANGE ROWS OF "XK" AND PUT INTO "XTEMP":
C---
      DO 450 I=1,6
      DO 450 J=1,6
      K=INDEX(I)
  450 XTEMP(I,J)=XK(K,J)
C---
C--- REARRANGE COLUMNS OF "XTEMP" AND PUT INTO "XK":
```

```
C---
      DO 460 I=1,6
      DO 460 J=1,6
      K=INDEX(I)
  460 XK(J,I)=XTEMP(J,K)
C---
C--- PUT "XK" ELEMENT MATRIX INTO BANDED FORM GLOBAL MATRIX "GK":
C---
      DO 470 I=1,6
      IROW=INDEM(I)
      II=IROW-1
      DO 470 J=1,6
      ICOL=INDEM(J)-II
  470 GK(IROW,ICOL)=GK(IROW,ICOL)+XK(I,J)
C---
C---
C--- SUBSTITUTE ELEMENT "XC" MATRIX INTO GLOBAL MATRIX "GC".
C---
C---
C--- REARRANGE ROWS AND COLUMNS OF ELEMENT "XC" MATRIX
C---
C--- REARRANGE ROWS OF "XC" AND PUT INTO "XTEMP":
C---
      DO 500 I=1,6
      DO 500 J=1,6
      K=INDEX(I)
  500 XTEMP(I,J)=XC(K,J)
C---
C--- REARRANGE COLUMNS OF "XTEMP" AND PUT INTO "XC":
C---
      DO 510 I=1,6
      DO 510 J=1,6
      K=INDEX(I)
  510 XC(J,I)=XTEMP(J,K)
C---
C--- PUT ELEMENT "XC" MATRIX INTO GLOBAL "GC" MATRIX.
C---
      DO 520 I=1,6
      IROW=INDEM(I)
      II=IROW-1
      DO 520 J=1,6
      ICOL=INDEM(J)-II
  520 GC(IROW,ICOL)=GC(IROW,ICOL)+XC(I,J)
      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COMP(Y,NROW,NCOL)
C---
C---
C---THIS SUBROUTINE MULTIPLIES BANDED SYMMETRIC MATRIX "GC" BY VECTOR
C---"Y" AND PUTS THE RESULTING VECTOR INTO COMMON VECTOR "Z".
C---
C---
      COMMON/BLK 2/GC(100,45)
      COMMON/BLK 3/Z(100)
      DIMENSION Y(100)
      DO 200 I=1,NROW
      Z(I)=Y(I)*GC(1,1)
      DO 200 K=2,NCOL
      L=I-K+1
      IF(L.LT.1) GO TO 100
      Z(I)=Z(I)+Y(L)*GC(L,K)
  100 N=I+K-1
      IF(N.GT.NROW) GO TO 200
      Z(I)=Z(I)+Y(N)*GC(I,K)
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FINSOL(NROW,NCOL)
C---
C---
C---THIS SUBROUTINE SOLVES A SET OF LINEAR SIMULTANEOUS
C---EQUATIONS WHOSE COEFFICIENT MATRIX HAS BEEN PRE-TRIANGULARIZED
C---BY THE GAUSSIAN ELIMINATION METHOD. THE SYSTEM MATRIX IS IN BAND
C---FORM, AND IS SYMMETRICAL. SOLUTION VECTOR IS PUT INTO COMMON VECTOR
C---"Z".
C---
      COMMON/BLK 1/A(100,40)
      COMMON/BLK 3/SS(100)
      COMMON/BLK 4/ST(100,40)
C---
C---REDUCE LOAD VARIABLES
C---
      N=0
  100 N=N+1
      SS(N)=SS(N)/A(N,1)
      IF (N-NRUN)  110,200,110
  110 CONTINUE
      DO 130  K=2,NCOL
       L=N+K-1
      IF(NROW-L) 130,120,120
  120 SS(L)= SS(L)-ST(N,K-1) *SS(N)
  130 CONTINUE
      GO TO 100
C---
C---  BACK SUBSTITUTION
C---
  200 N=NROW
  300 N=N-1
      IF (N) 350,500,350
  350 DO 400 K=2, NCOL
      L=N+K-1
      IF (NROW-L) 400,370,370
  370 SS(N)= SS(N)- A(N,K) * SS(L)
  400 CONTINUE
      GO TO 300
  500 RETURN
      END
```

```
      SUBROUTINE PRESOL (NROW, NCOL)
C---
C---
C---   THIS SUBROUTINE PRE TRIANGULARIZES A SYMMETRIC
C---   MATRIC IN BAND FORM FOR SOLUTION BY THE GAUSSIAN
C---ELIMINATION METHOD. FINAL SOLUTION IS BY SUBROUTINE FINSOL.
C---
C---
      COMMON/BLK 1/W(100,40)
      COMMON/BLK 4/ST(100,40)
      N=0
  100 N=N+1
C---
C---   REDUCE PIVOT EQUATION
C---
      IF (N-NROW) 150,300,150
  150 DO 200 K=2, NCOL
      ST (N,K-1)= W(N,K)
  200 W(N,K)= W(N,K) /W(N,1)
C---
C----  REDUCE REMAINING EQUATIONS
C---
      DO 260  L=2,NCOL
      I=N+L-1
      IF(NROW-I)260,240,240
  240  J=0
      DO 250  K=L,NCOL
      J=J+1
  250 W(I,J)=W(I,J)-ST(N,L-1) *W(N,K)
  260 CONTINUE
      GO TO 100
  300  RETURN
      END
```