Finite Element Analysis of Nonlinear Heat Transfer Problems

Polivka, Ronald
Wilson, Edward

Publication Date
1976-06-01

REPORT NO.
UCB/SESM-76/2

STRUCTURAL ENGINEERING AND
STRUCTURAL MECHANICS

DOT/DETECT
FINITE ELEMENT ANALYSIS OF NONLINEAR HEAT TRANSFER PROBLEMS

by
RONALD M. POLIVKA
AND
EDWARD L. WILSON

A Computer Program
Distributed by
National Information Service
for Earthquake Engineering

JUNE 1976

DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA

FINITE ELEMENT ANALYSIS OF NONLINEAR HEAT TRANSFER PROBLEMS

by

Ronald M. Polivka

and

Edward L. Wilson

June, 1976

Structural Engineering and Structural Mechanics
Department of Civil Engineering
University of California, Berkeley

ACKNOWLEDGMENT

The research described in this report was submitted in partial fulfillment of the requirements for the Ph.D. degree in Engineering at the University of California, Berkeley.

The author wishes to express his sincere appreciation to Professor Edward L. Wilson for his guidance and support throughout the course of this research. He is also most grateful to Professors Oscar E. Lanford III and David Pirtz for serving as members of his thesis committee.

The author wishes to express his gratitude to Fred Peterson for his valuable suggestions during the development of this work. Thanks are also due to Digambar Mondkar for many helpful suggestions.

The author is indebted to his father, Professor Milos Polivka, and mother, Joyce for their assistance in reviewing the manuscript.

Finally, I would like to thank my wife Victoria for her encouragement and understanding.

TABLE OF CONTENTS

Page

Page

1. INTRODUCTION

The finite element method has gained widespread use for the stress analysis of complex structural systems. In previous applications of the method, the complete thermal stress analysis of most structures involved two separate phases. First, the heat transfer problem alone is solved based on certain idealizations, and then the resulting temperature distribution is used in connection with certain structural idealizations to determine the thermal stresses within a structure. Because different idealizations are used for both the heat conduction and thermal stress analyses, a separate computer input must be prepared for each phase. It is desirable, therefore, to develop compatible heat transfer and stress analysis programs in order to minimize the preparation of data required for the thermal stress problem.

In the past decade, several approximate solution techniques for heat conduction problems have been developed based on the finite element method. The method was first applied by Zienkiewicz and Cheung [1] and Wilson [2] for the solution of steady state conduction problems and extended by Wilson and Nickell [3] to include linear transient problems. In Ref. [3], the method was developed in detail for two dimensional planar bodies idealized by a system of triangular finite elements.

It is widely known that the thermal conductivity and specific heat capacity of many materials can vary with temperature. Few attempts have been made, however, to assess the importance of this temperature-dependence because it leads to considerable nonlinearities in the governing equations for heat flow equilibrium.

This dissertation is concerned with the development of procedures for the solution of nonlinear transient heat conduction problems by the finite element method. Although the method has been developed for two-dimensional structures, it can be readily extended to the heat conduction analysis of three dimensional systems. A variable 4- to 8-node isoparametric element is used in the analysis. This was done to provide compatibility with some of the more recently developed stress analysis programs.

The finite element method is completely general with respect to geometry, material properties and boundary conditions. Complex bodies composed of several different anisotropic materials can be easily represented. Temperature or heat flux boundary conditions may be specified at any point within the finite element system. Additionally, the finite element approach generates a heat flow equilibrium equation for the system which contains a symmetric positive-definite matrix. This matrix is banded, and the equilibrium equations can therefore be solved with a minimum of computer storage and time.

The following sections contain the theoretical basis for the solution of nonlinear transient heat conduction problems. Based on this theory, two finite element analysis programs have been developed. The program DOT ("Determination Of Temperatures") is a general purpose heat transfer analysis program for both linear and nonlinear conduction problems, and the program DETECT ("DEtermination of TEmperatures in ConsTruction") is a linear heat transfer program for the analysis of structures constructed incrementally. A description of the required input data for these programs is given in Appendices A and B, respectively, along with the Fortran listings.

2. HEAT TRANSFER ANALYSIS

2.1 EQUATIONS FOR EQUILIBRIUM

For the purpose of illustration, the heat transfer equations will be presented for a two-dimensional problem. The analysis can, however, easily be extended to three dimensions. The nonlinear two-dimensional equation for transient heat conduction in an anisotropic and inhomogenous material is given by:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + Q \qquad (2.1)$$

where

T = unknown temperature

Q = heat generation rate

$\left.\begin{matrix} k_x \\ k_y \end{matrix}\right\}$ = anisotropic conductivity coefficients (temperature and space dependent)

ρ = mass density (space dependent)

c = specific heat capacity (temperature and space dependent)

x,y = rectangular cartesian coordinates

t = time

If the thermal conductivity is constant and the material is isotropic, then $k_x = k_y = k$ and Eq. (2.1) reduces to the more familiar form:

$$\nabla^2 T + \frac{Q}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \qquad (2.2)$$

where the thermal diffusivity $\alpha = k/\rho c$ and ∇ is the gradient operator.

In order to determine a unique solution to Eq. (2.1), proper initial and boundary conditions must be specified which are compatible with the physical conditions of the particular problem.

2.2 INITIAL CONDITION

The initial condition must be defined by prescribing the temperature distribution throughout the body at time zero as a known function of x and y:

$$T(x,y,0) = f(x,y) \quad @ \quad t = 0 \tag{2.3}$$

2.3 BOUNDARY CONDITIONS

Boundary conditions may be defined by specifying either the prescribed surface temperature or heat flow on the boundary of the body. In many problems, heat may also be transferred by convection or radiation across a boundary layer and these boundary conditions must also be specified.

2.3.1 Prescribed Surface Temperature

A surface temperature existing on a portion of the boundary of the body Γ_T must be specified as:

$$T(x,y,t) = T_1(x,y,t) \quad ; \quad x,y \text{ on } \Gamma_T \tag{2.4}$$

This is the easiest boundary condition to work with and the temperature may be either constant or a function of time.

2.3.2 Prescribed Heat Flow at the Surface

A prescribed heat flow boundary condition can be expressed as:

$$k\,\frac{\partial T}{\partial n}(x,y,t) = q_n(x,y,t) \quad ; \quad x,y \text{ on } \Gamma_q \tag{2.5}$$

where q_n is the specified amount of heat flow at point (x,y) on that portion Γ_q of the boundary of the body where the normal derivative of temperature is prescribed, n is the outward normal to this surface and k is the thermal conductivity.

2.3.3 Convection Boundary Condition

For the case of free convection, the rate of heat transfer across a boundary layer is given by:

$$k \frac{\partial T}{\partial n}(x,y,t) \quad = \quad h(T_e - T_s)^N \quad ; \quad x,y \text{ on } \Gamma_h \qquad (2.6)$$

where h is the surface heat transfer coefficient, T_e is the known temperature of the external environment, T_s is the surface temperature of the solid and Γ_h is that portion of the boundary surface undergoing convective heat transfer.

The convection boundary condition is linear and can be specified exactly if N = 1 and h is not a function of temperature. If, in addition, there is a prescribed flux $q_n(x,y,t)$ at the surface, Eq. (2.6) is replaced by:

$$k \frac{\partial T}{\partial n}(x,y,t) \quad = \quad q_n(x,y,t) \quad + \quad h(T_e - T_s) \qquad (2.7)$$

2.3.4 Radiation Boundary Condition

Heat transfer by radiation between a boundary surface Γ_r and its surroundings can be expressed by:

$$q_r(x,y,t) \quad = \quad V\sigma \left[\frac{1}{\frac{1}{\varepsilon_r} + \frac{1}{\varepsilon_s} - 1} \right] [\bar{T}_r^4 - \bar{T}_s^4] \quad ; \quad x,y \text{ on } \Gamma_r \qquad (2.8)$$

where V is the radiation view factor, σ the Stefan-Boltzmann constant, ε_r is the emissivity of the external radiation source, ε_s is the emissivity of the surface, and $\bar{T}_r$ and $\bar{T}_s$ are the absolute temperatures of the radiation source and the surface, respectively.

2.4 SOLUTION OF THE HEAT FLOW EQUILIBRIUM EQUATION

Three types of methods are available for solving the general heat transfer equation (Eq. (2.1)) for the temperature distribution within a body: exact methods, approximate analytical methods and approximate numerical methods.

2.4.1 Exact Solutions

The standard classical reference on heat transfer analysis is the book by Carslaw and Jaeger [4], in which a number of exact solutions are given for rectangular regions, infinite and semi-infinite solids bounded by two parallel planes, cylinders and spheres that are subjected to a variety of initial and boundary conditions. The methods used for these exact solutions range from simple techniques such as separation of variables to complex ones involving the use of Green's functions, Laplace transforms, integral transforms or Fourier transforms.

In almost all of the problems for which an exact solution is possible, the thermal properties k, ρ and c are taken to be constants, independent of both temperature and position. If the thermal properties depend on temperature, the solution to Eq. (2.1) is much more complicated since the equation becomes nonlinear. In the majority of these cases, no exact solution exists, and the temperature field must be determined by numerical methods.

2.4.2 Analytical Solutions

This approximate method of solution requires that an initial temperature profile be assumed before the solution can be carried out. Two of the more commonly used techniques are the Laplace-variational method [5] and the integral method [6].

Analytical methods of solution are capable of solving a wider variety of heat conduction problems than can be solved by exact methods; e.g. a one dimensional problem with temperature-dependent thermal conductivity.

2.4.3 Numerical Methods

The most commonly used numerical method for solving the heat equation (Eq. (2.1)) is the finite difference method [7]. In this implicit method, a truncated Taylor series expansion is used to approximate the derivatives in the governing differential equation and the space and time derivatives are then replaced by a finite difference representation. The resulting temperature distribution can then be obtained by solving a set of simultaneous algebraic equations at each level of time.

In recent years, the finite element method has gained widespread use in the stress analysis of complex structural problems. For the evaluation of thermal stresses, the finite element technique is desirable for solving the heat flow problem because the same model can be used for both the stress and heat transfer analyses, thus reducing the required amount of input data.

It is customarily assumed that the state of deformation in a body does not affect the temperature field. Therefore, a heat transfer analysis can be conducted initially using a finite element idealization and then a second pass can be made through the same program to determine the thermal stresses. If the initial heat transfer analysis had been performed using some other numerical technique, a special thermal analyzer would have been required to determine the temperature input compatible with the finite element stress program.

3. FINITE ELEMENT FORMULATION

In the finite element method of analysis, a solid continuum is idealized by an assemblage of discrete elements or subregions. These elements may be of variable size and shape, and are interconnected by a finite number of nodal points. The element boundaries are generally linear, although recent development of the 4- to 8-node isoparametric element has permitted the use of curved boundaries.

For the case of heat conduction, the temperature field within each element is approximated by a set of interpolation functions $\underline{h}_i$, chosen in such a way so as to define the temperature uniquely within each finite element in terms of its nodal temperatures. The matrix equations for heat flow equilibrium can then be derived for the discrete number of nodal points within the finite element system.

The finite element method has several advantages over other solution procedures. The method is completely general with respect to geometry, material properties and arbitrary boundary conditions. Complex bodies of arbitrary shape composed of several different anisotropic materials can easily be represented. A mixture of temperature-dependent material properties and nonlinear boundary conditions can also be handled quite readily. The governing heat flow equilibrium equations for the response of the discrete system produce a symmetric, positive-definite matrix which may be placed in a band form. Efficient solution techniques can therefore be applied which require a minimum amount of computer time and storage.

Initially, the basic element types used to represent a two-dimensional body were triangular and rectangular elements containing straight-line boundaries. However, the recent development of the variable 4- to 8-node isoparametric element has given the method much more flexibility [8]. This element is very useful for representing complex bodies with curved boundaries and it readily allows for varying the mesh size within a model.

A variable 4- to 8-node two-dimensional isoparametric element lying in the global X-Y plane is shown in Fig. 3.1(b). This element may be used to model either planar or axisymmetric solids. The planar element is assumed to have a unit thickness, whereas in the axisymmetric representation a unit radian segment ($\theta = 1$) is considered, with the global Y-axis as the axis of revolution.

This curvilinear 4- to 8-node element can be obtained by means of an isoparametric mapping from a bi-unit square contained in a local r-s coordinate system. The local node numbering system for this bi-unit square, shown in Fig. 3.1(a), has nodes 1 through 4 located at the four corners and nodes 5 through 8 located at the midsides of the square.

Isoparametric mapping provides a one-to-one correspondence between the local (r,s) and global (x,y) coordinates. The coordinate transformation between the bi-unit square and the curvilinear element is given by:

$$x_m(r,s) = \sum_{i=1}^{8} h_{im}(r,s)\, x_{im} \tag{3.1}$$

$$y_m(r,s) = \sum_{i=1}^{8} h_{im}(r,s)\, y_{im} \tag{3.2}$$

where (x_{im},y_{im}) are the global coordinates of node i in element m and h_{im} is the interpolation function corresponding to node i of element m. The interpolation functions are defined as follows:

$$\begin{aligned}
h_1 &= \frac{1}{4}(1 + r)(1 + s) - \frac{1}{2}h_5 - \frac{1}{2}h_8 \\
h_2 &= \frac{1}{4}(1 - r)(1 + s) - \frac{1}{2}h_5 - \frac{1}{2}h_6 \\
h_3 &= \frac{1}{4}(1 - r)(1 - s) - \frac{1}{2}h_6 - \frac{1}{2}h_7 \\
h_4 &= \frac{1}{4}(1 + r)(1 - s) - \frac{1}{2}h_7 - \frac{1}{2}h_8 \qquad (3.3) \\
h_5 &= \frac{1}{2}(1 - r^2)(1 + s) \\
h_6 &= \frac{1}{2}(1 - r)(1 - s^2) \\
h_7 &= \frac{1}{2}(1 - r^2)(1 - s) \\
h_8 &= \frac{1}{2}(1 + r)(1 - s^2)
\end{aligned}$$

where the local coordinates (r,s) vary on the interval (-1,1). If the curvilinear element has one or more straight sides, the midside node numbers 5, 6, 7 or 8 corresponding to the straight sides can be omitted by setting the corresponding interpolation functions equal to zero.

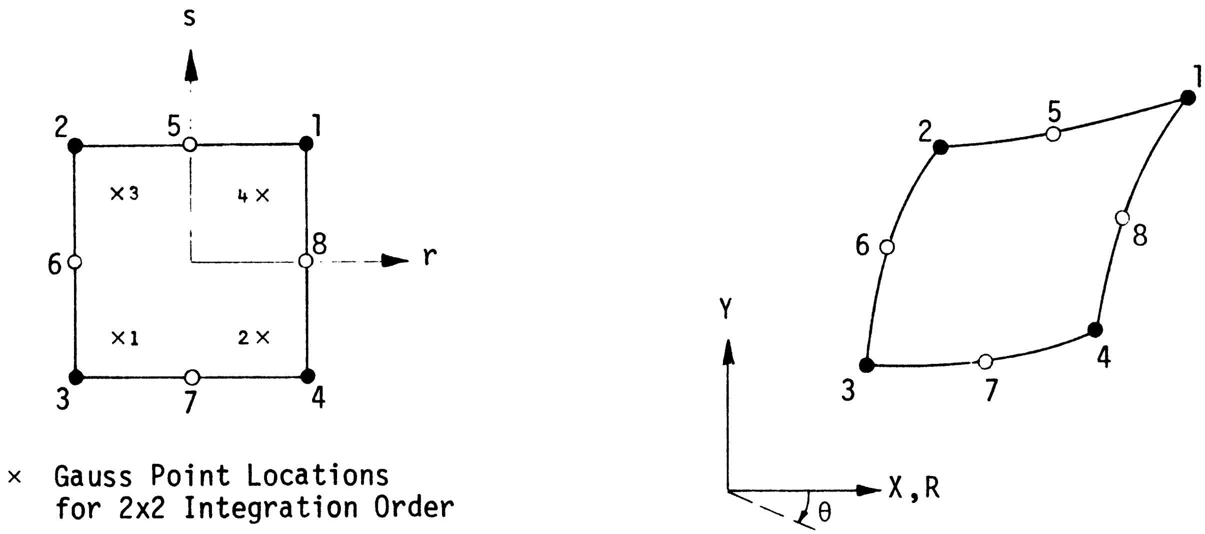
The same interpolation functions are used to approximate the temperature field within the element in terms of the temperatures at nodes 1 to 8:

$$T_m(r,s,t) = \sum_{i=1}^{8} h_{im}(r,s)\, T_{im}(t) \qquad (3.4)$$

where T_{im} is the temperature of the i^{th} nodal point of element m. Writing in matrix notation we have:

$$T_m(r,s,t) = \underset{1\times 8}{\underset{\sim}{b}_m(r,s)}\ \underset{8\times 1}{\underset{\sim}{T}_m(t)} \qquad (3.5)$$

where $\underset{\sim}{b}_m(r,s)$ is the transformation matrix between the nodal point temperatures $\underset{\sim}{T}_m(t)$ and the temperature $\underset{\sim}{T}_m(r,s,t)$ at any point (r,s) within element m.



× Gauss Point Locations
for 2x2 Integration Order

(a) Bi-unit Square in
Local r-s System

(b) Curvilinear Two-Dimensional
Element in Global X-Y System

FIG. 3.1 TWO-DIMENSIONAL MAPPING OF 4- TO 8-NODE ISOPARAMETRIC ELEMENT

In the evaluation of the element conductivity matrix $\underline{K}_m$, defined in the following section, it is required to know the transformation between the nodal temperatures and the global derivatives of the element temperature. This transformation is analogous to the strain-displacement transformation required in a stress analysis problem.

The transformation between the nodal temperatures and the local derivatives of the element temperature can be found by direct differentiation of Eq. (3.5):

$$\frac{\partial}{\partial r} T_m(r,s,t) = \sum_{i=1}^{8} \frac{\partial}{\partial r} h_{im}(r,s)\; T_{im}(t) \qquad (3.6)$$

$$\frac{\partial}{\partial s} T_m(r,s,t) = \sum_{i=1}^{8} \frac{\partial}{\partial s} h_{im}(r,s)\; T_{im}(t) \qquad (3.7)$$

Combining the above two equations and writing in matrix form we have:

$$\begin{bmatrix} \frac{\partial T}{\partial r} \\ \frac{\partial T}{\partial s} \end{bmatrix}_m = \underset{2\times 8}{\underline{P}_m(r,s)}\; \underset{8\times 1}{\underline{T}_m(t)} \qquad (3.8)$$

where $\underline{P}_m(r,s)$ contains the derivatives of the interpolation functions for element m with respect to the local (r,s) coordinates. The chain rule can now be applied to obtain these derivatives with respect to the global (x,y) coordinates. The chain rule may be written in matrix form as:

$$\begin{bmatrix} \frac{\partial T}{\partial r} \\ \frac{\partial T}{\partial s} \end{bmatrix}_m = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix}_m \qquad (3.9)$$

and inverted as

$$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix}_m = \underline{J}_m^* \begin{bmatrix} \frac{\partial T}{\partial r} \\ \frac{\partial T}{\partial s} \end{bmatrix}_m \tag{3.10}$$

where

$$\underline{J}_m^* = \frac{1}{|\underline{J}|_m} \begin{bmatrix} \frac{\partial y}{\partial s} & -\frac{\partial y}{\partial r} \\ -\frac{\partial x}{\partial s} & \frac{\partial x}{\partial r} \end{bmatrix} \tag{3.11}$$

in which the determinant of the Jacobian transformation is

$$|\underline{J}|_m = \frac{\partial x}{\partial r}\frac{\partial y}{\partial s} - \frac{\partial y}{\partial r}\frac{\partial x}{\partial s} \tag{3.12}$$

The transformation between the nodal temperatures and the global derivatives of the element temperature can now be found by substituting Eq. (3.8) into the inverted chain rule Eq. (3.10):

$$\begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix}_m = \underline{J}_m^* \underline{P}_m \underline{T}_m(t) \tag{3.13}$$

or

$$\underline{T}_{,i}(r,s,t) = \underset{2\times 8}{\underline{a}_m(r,s)} \; \underset{8\times 1}{\underline{T}_m(t)} \tag{3.14}$$

3.1 MATRIX EQUILIBRIUM EQUATIONS FOR TRANSIENT HEAT CONDUCTION

The governing equations for transient heat conduction in a solid idealized by a system of finite elements may be derived in a number of ways. In one approach, a functional is defined by some integral over the entire domain and its boundary, in which the unknown temperature field or its derivatives appear. The temperature field providing the correct solution to the problem is the one that causes the functional to be minimized. From this minimization of the functional results the governing partial differential equations.

This class of solution techniques, often referred to as variational methods [1,3], [9-19] is the most widely accepted technique for arriving at a finite element representation of the equilibrium equations.

A second approach requires prior knowledge of the differential equations governing the problem. The finite element approximation can then be directly derived mathematically from the differential equations by the method of weighted residuals [20,21]. Depending upon the choice of the weighting function, different techniques can be used such as the collocation method or the Galerkin method. The finite element derivation using the variational formulation is a special case of the method of weighted residuals using the Galerkin method for the weighting function.

A third method involves a purely physical interpretation of the equations for heat flow equilibrium [22-25]. This method has the advantage of providing insight to the solution of nonlinear analysis problems and is the one that will be used in this presentation.

3.1.1 Physical Statement of Heat Flow Equilibrium

The heat transfer equations for a solid idealized by a system of finite elements can be physically interpreted by a statement of the heat flow equilibrium at the nodes of the system at any time:

$$Q^s + Q^i = Q^e + Q^q \qquad (3.15)$$

where

Q^s = rate at which heat is stored within the elements adjacent to a node

Q^i = rate of heat flow from the elements adjacent to a node by means of internal heat conduction

Q^e = rate at which heat enters a node from an external source

Q^q = rate at which heat is generated within the elements adjacent to a node

This heat balance equation is valid for all interior nodal points, but must be modified for nodes on the boundary surfaces in order to account for a variety of boundary conditions which will be described later.

Using the thermal properties of the finite elements, the heat transfer rates in Eq. (3.15) can be expressed in terms of the temperatures at the element nodes.

The instantaneous rate at which heat is stored within the material at any time t is:

$$q^s = \rho c(T) V \frac{\partial T}{\partial t} = C(T) \frac{\partial T}{\partial t} \qquad (3.16)$$

where the thermal heat capacity C of the material is a function of the density ρ, the volume V and the temperature-dependent specific heat capacity c of the material.

If the heat flow is defined as positive for a positive temperature gradient, the rate of heat conduction per unit area along direction j within the material can be expressed as:

$$q_j^i = \sum_n k_{jn}(T) \frac{\partial T}{\partial n} \tag{3.17}$$

where k_{jn} is a symmetric positive-definite thermal conductivity tensor for the material which may vary with temperature, and n indicates the direction of the temperature gradient.

For a two-dimensional problem with coordinates x and y, Eq. (3.17) can be expressed in matrix form as:

$$\begin{bmatrix} q_x^i \\ q_y^i \end{bmatrix} = \begin{bmatrix} k_{xx} & k_{xy} \\ k_{xy} & k_{yy} \end{bmatrix} \begin{bmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \end{bmatrix} \tag{3.18}$$

where the three constants k_{xx}, k_{xy} and k_{yy} which characterize the tensor must satisfy the conditions:

$$\begin{aligned} &(a) \quad k_{xx}, k_{yy} > 0 \\ &(b) \quad k_{xx}k_{yy} - k_{xy}^2 > 0 \end{aligned} \tag{3.19}$$

For an isotropic material, $k_{xx} = k_{yy} = k$ and $k_{xy} = 0$.

Using the thermal properties defined in Eqs. (3.16) and (3.17) along with the interpolation functions for the element temperature given in Eqs. (3.5) and (3.14), the method of virtual work can be applied in order to express the heat balance rates in Eq. (3.15) in terms of the nodal point temperatures [26].

The resulting matrix differential equation governing heat flow equilibrium in a finite element system containing n nodes is:

$$\underset{n\times n}{\underline{C}(T)}\;\underset{n\times 1}{\dot{\underline{T}}(t)} + \underset{n\times n}{\underline{K}(T)}\;\underset{n\times 1}{\underline{T}(t)} = \underset{n\times 1}{\underline{Q}(t)} \qquad (3.20)$$

where

$\underline{C}(T)$ = system heat capacity matrix which is dependent on the density ρ and the temperature-dependent specific heat capacity c of the material

$\underline{K}(T)$ = system thermal conductivity matrix which is dependent on the temperature T of the body

$\underline{T}(t)$ = vector of nodal point temperatures at time t

$\dot{\underline{T}}(t)$ = vector of the time rate-of-change of the nodal point temperatures at time t

$\underline{Q}(t)$ = vector of the externally supplied nodal heat fluxes at time t (heat which is generated within the elements can be considered in this vector)

Equation (3.20) represents a set of nonlinear equations to be solved for the nodal point temperatures of the finite element representation as a function of time. The matrices $\underline{C}$, $\underline{K}$ and $\underline{Q}$ are known at any instant in time and Eq. (3.20) can therefore be solved using numerical integration.

In the following sections, the matrices $\underline{K}$, $\underline{C}$ and $\underline{Q}$ are formulated and the numerical integration algorithm used to solve Eq. (3.20) will be described in Section 5.

3.2 SYSTEM MATRICES

3.2.1 Conductivity Matrix

The complete conductivity matrix for the system is formed by direct summation of the element conductivity matrices for all of the element types. This assembly operation is analogous to the direct stiffness assembly procedure and is indicated symbolically by means of the following equation:

$$\underline{K} = \sum_{m=1}^{M} \underline{K}_m \qquad (3.21)$$

where M is the total number of elements in the finite element model.

The matrix $\underline{K}_m$ is the conductivity matrix of element m and is defined as:

$$\underline{K}_m = \int_{V_m} \underline{a}_m^T \, \underline{k}_m \, \underline{a}_m \, dV_m \qquad (3.22)$$

where

$\underline{k}_m$ = thermal conductivity tensor for the material (from Eq. (3.18))

$\underline{a}_m$ = transformation between the nodal temperatures and the global derivatives of the element temperature given by Eq. (3.14)

V_m = volume of element m

Since the transformation matrix $\underline{a}_m$ is expressed in terms of the natural coordinates r and s, it is necessary to carry out the above integration in natural coordinates.

(a) Planar Solid

For a planar solid of constant thickness t^*, Eq. (3.22) can be expressed as:

$$\underset{\sim}{K}_m = t^* \int_{A_m} \underset{\sim}{a}_m^T \underset{\sim}{k}_m \underset{\sim}{a}_m \, dA_m \tag{3.23}$$

In the natural coordinate system the differential area is:

$$dA_m = dx\,dy = |\underline{J}|\,ds\,dt \tag{3.24}$$

where $|\underline{J}|$ is the determinant of the Jacobian matrix from Eq. (3.12).

Substituting Eq. (3.24) into (3.23) and assuming a unit thickness we have:

$$\underset{\sim}{K}_m = \int_{-1}^{1}\int_{-1}^{1} \underset{\sim}{a}_m^T \underset{\sim}{k}_m \underset{\sim}{a}_m \, |\underline{J}| \, dr\,ds \tag{3.25}$$

This is of the general form:

$$\underset{\sim}{K}_m = \int_{-1}^{1}\int_{-1}^{1} f(r,s)\,dr\,ds \tag{3.26}$$

and the integral Eq. (3.25) can now be evaluated numerically using Gaussian quadrature:

$$\underset{\sim}{K}_m = \sum_{j=1}^{N}\sum_{i=1}^{N} H_i H_j \, \underset{\sim}{a}_m^T(r_i,s_j) \, \underset{\sim}{k}_m \, \underset{\sim}{a}_m(r_i,s_j) \, |\underline{J}(r_i,s_j)| \tag{3.27}$$

where

H_i, H_j = weighting coefficients for Gaussian integration

r_i, s_j = abscissae of the integration points

N = integration order

The weight coefficients and abscissae for the Gaussian quadrature formula may be found in Ref. [26]. Integration point locations for an integration order of 2 are shown in Fig. 3.1(a). For the DOT and DETECT computer programs described in Appendices A and B, respectively, integration orders of either 2 or 3 may be used. It is recommended, however, to use a 2x2 quadrature integration scheme for most cases.

(b) Axisymmetric Solid

For an axisymmetric solid, the volume integral for the element conductivity can be converted to an area integral since dV = R dθ dA, where the radius R and the angle θ are defined in Fig. 3.1(b). For a one radian segment of the solid:

$$\underline{K}_m = \int_{A_m} R \; \underline{a}_m^T \; \underline{k}_m \; \underline{a}_m \; dA_m \tag{3.28}$$

In the natural coordinate system $dA_m = |\underline{J}|$ ds dt, therefore:

$$\underline{K}_m = \int_{-1}^{1}\int_{-1}^{1} R \; \underline{a}_m^T \; \underline{k}_m \; \underline{a}_m \; |\underline{J}| \; ds \; dt \tag{3.29}$$

The integral Eq. (3.29) can now be evaluated numerically using Gaussian quadrature as:

$$\underline{K}_m = \sum_{j=1}^{N}\sum_{i=1}^{N} H_i \; H_j \; \underline{f}(r_i,s_j) \tag{3.30}$$

where

$$\underline{f}(r_i,s_j) = \underline{a}_m^T(r_i,s_j) \; \underline{k}_m \; \underline{a}_m(r_i,s_j) \; R(r_i,s_j) \; |\underline{J}(r_i,s_j)| \tag{3.31}$$

3.2.2 Heat Capacity Matrix

The complete heat capacity matrix for the system is also formed by direct summation of the individual element heat capacity matrices:

$$\underline{C} = \sum_{m=1}^{M} \underline{C}_m \tag{3.32}$$

where M is the total number of finite elements in the model.

The heat capacity matrix $\underline{C}_m$ of element m is defined as:

$$\underline{C}_m = \int_{V_m} \rho_m \, c_m \, \underline{b}_m^T \, \underline{b}_m \, dV_m \tag{3.33}$$

where

c_m = specific heat capacity

ρ_m = mass density

$\underline{b}_m$ = temperature interpolation vector given by Eq. (3.5)

V_m = volume of element m

The computation of $\underline{C}_m$ in Eq. (3.33) results in a fully populated matrix identical in form to the thermal conductivity matrix $\underline{K}_m$. The assembled heat capacity matrix $\underline{C}$ is symmetric, positive-definite, and exhibits the same coupling as the system conductivity matrix $\underline{K}$. This coupled $\underline{C}$ matrix increases significantly the computational effort required to solve the heat flow equilibrium equation (3.20). It has been shown, however, that the element heat capacity matrix $\underline{C}_m$ can be approximated by a lumped diagonal matrix with only a small loss of accuracy [3,22]. This approximation reduces by almost 50% the computer storage and computational steps required in the solution of Eq. (3.20).

The concept of lumping the heat capacity of an element at the nodes used to describe the element is equivalent to the lumped mass idealization in dynamic analysis. This lumping eliminates any coupling between the time rate-of-change of temperature at adjacent nodes and results in a diagonal heat capacity matrix $\underset{\sim}{C}_m$. The accuracy of the lumping technique used depends on the number of nodes used to describe the element and the element shape.

(a) 4-Node Elements

For a regular finite element grid comprised of either 3- or 4-node elements, the lumped heat capacity matrix can be formed as:

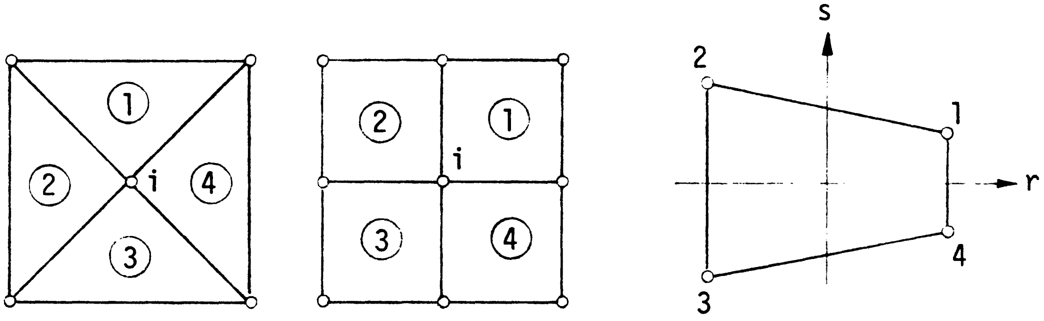
$$C_{ii} = \sum_m \frac{1}{M} \rho_m c_m V_m \tag{3.34}$$

where the summation is carried out over all m elements attached to nodal point i. The parameter M is the number of nodes describing each element, and the density ρ_m, specific heat c_m and volume V_m are those values associated with each element m (See Fig. 3.2(a)). This lumping procedure has been used in several earlier heat transfer programs [3,22] and stress analysis programs [27,28] involving 3- to 4-node elements. Inaccuracies can arise, however, if the grid has distorted elements. For the 4-node element shown in Fig. 3.2(b), the heat capacity would be lumped evenly to all nodes even though nodes 2 and 3 should be weighed heavier since they have a larger tributary volume.

For a distorted finite element grid, therefore, a more accurate formula to use in lumping the system heat capacity matrix is:

$$C_{ii} = \sum_m \sum_{j=1}^{N} \int_{V_m} \rho_m c_m h_{jm} dV_m \tag{3.35}$$

where m is the number of elements attached to node i, N is the number of nodes in each element and h_{jm} is the interpolation function corresponding to node j of element m. For any element m, the integral in Eq. (3.35) is exactly equal to $\rho_m c_m V_m$ for the element. Equation (3.35) is not applicable, however, to higher order elements because negative values of C_{ii} will result if the midside nodes are close to the corner nodes.



(a) Regular Grids (b) Distorted Element

FIG. 3.2 HEAT CAPACITY IDEALIZATION FOR 4-NODE ELEMENTS

(b) Higher Order Elements

The DOT and DETECT finite element programs described in Appendices A and B employ a higher order 4- to 8-node element. For these programs, it was necessary to use a lumped heat capacity approximation for the consistent element heat capacity matrix, defined as:

$$(C_{ij})_m = \int_{V_m} \rho_m \, c_m \, h_{im} \, h_{jm} \, dV_m \tag{3.36}$$

The steps used for lumping the element heat capacity matrix for higher order elements are:

(1) Calculate diagonal terms from consistent heat capacity equation (3.36):

$$(C_{ii})_m = \int_{V_m} \rho_m \, c_m \, h_{im}^2 \, dV_m \tag{3.37}$$

(2) Compute the total heat capacity of element m:

$$(C_T)_m = \int_{V_m} \rho_m \, c_m \, dV_m \tag{3.38}$$

(3) Compute sum of diagonal terms:

$$S_m = \sum_{i=1}^{N} (C_{ii})_m \cong (C_T)_m \tag{3.39}$$

where N = number of nodes in element m.
The sum S_m is approximately equal to $(C_T)_m$.

(4) Apply corrections to the diagonal terms $(C_{ii})_m$ so that $S_m \equiv (C_T)_m$:

$$(C_{ii}^{*})_m = R_m (C_{ii})_m \tag{3.40}$$

where

$$R_m = \frac{(C_T)_m}{S_m} \quad ; \text{ correction factor}$$

$(C_{ii}^{*})_m$ = lumped diagonal heat capacity terms for element m.

All of the integrations in the above steps are computed using Gaussian quadrature, and the lumped element heat capacity matrices resulting from Eq. (3.40) are assembled into the system heat capacity matrix $\underline{C}$ by the direct summation shown in Eq. (3.32).

3.2.3 Heat Flux Vector

The system heat flux vector $\underset{\sim}{Q}(t)$ at time t is made up of the contributions from four separate sources:

$$\underset{\sim}{Q}(t) = \underset{\sim}{Q}^1(t) + \underset{\sim}{Q}^2(t) + \underset{\sim}{Q}^3(t) + \underset{\sim}{Q}^4(t) \qquad (3.41)$$

where $\underset{\sim}{Q}^1(t)$ = externally supplied nodal heat flux (time dependent)

$\underset{\sim}{Q}^2(t)$ = convective heat transfer at the surface (time and temperature dependent)

$\underset{\sim}{Q}^3(t)$ = radiative heat transfer at the surface (time and temperature dependent)

$\underset{\sim}{Q}^4(t)$ = internal heat generation (time dependent)

A. Nodal Heat Fluxes

In the finite element solution of heat conduction problems, either a temperature or an externally supplied heat flux must be associated with each node. The external heat flow Q_i supplied to node i can either be constant or a prescribed function of time, and the system heat flux vector $\underset{\sim}{Q}^1(t)$ is formed by direct assembly of these Q_i contributions.

B. Convective Heat Transfer

In the case of linear free convection, the rate of heat flux across a boundary layer at the surface of a body per unit area is:

$$q^2 = h^c \, [T_e - T_s] \qquad (3.42)$$

where h^c = heat transfer coefficient for the boundary layer which may be time dependent.

T_e = temperature of the external environment which is a known function of time.

T_s = unknown surface temperature of the body.

Assume that a boundary element m lies between nodes i and j and is subjected to surface heat transfer of the form described by Eq. (3.42). The surface temperature field for the boundary element can be expressed in terms of the nodal temperatures by the approximate relationship:

$$T_m(\bar{x},t) = \underset{1\times 2}{\underset{\sim}{\phi}_m(\bar{x})} \; \underset{2\times 1}{\underset{\sim}{T}_m(t)} \qquad (3.43)$$

where

$\underset{\sim}{\phi}_m(\bar{x})$ = interpolation vector describing the variation of temperature along boundary element m.

$\underset{\sim}{T}_m(t)$ = nodal point temperature vector for boundary element m.

$\bar{x}$ = spatial coordinate

The contribution to the surface heat flux vector $\underset{\sim}{Q}^2(t)$ at any time t is given by the surface integral [3,26]:

$$\underset{2\times 1}{\underset{\sim}{q}^2_m} = \int_{S_m} h^c_m \, T_e \, \underset{2\times 1}{\underset{\sim}{\phi}^T_m} \, dS_m \qquad (3.44)$$

where

h^c_m = heat transfer coefficient for boundary element m.

S_m = surface area of boundary element m.

$\underset{\sim}{\phi}_m$ = temperature interpolation vector given by Eq. (3.43).

The heat flux components in $\underset{\sim}{q}^2_m$ are added directly into the heat flux vector $\underset{\sim}{Q}^2(t)$.

For convective heat transfer, the appropriate terms of the system conductivity matrix $\underset{\sim}{K}$ must be modified by the coefficients of the following matrix:

$$\underset{2\times 2}{\underset{\sim}{K}^2_m} = \int_{S_m} h^c_m \, \underset{\sim}{\phi}^T_m \, \underset{\sim}{\phi}_m \, dS_m \qquad (3.45)$$

The matrix $\underset{\sim}{K}_m^2$ is directly assembled into $\underline{K}$ for nodes i and j which define the boundary element m.

The matrices $\underset{\sim}{q}_m^2$ from Eq. (3.44) and $\underset{\sim}{K}_m^2$ from Eq. (3.45) will now be derived for both planar and axisymmetric solids.

(a) Planar Solid

Consider the convection boundary element m in Fig. 3.3(a) which lies between nodes i and j on the surface of a planar solid. The temperature interpolation vector $\underset{\sim}{\phi}_m$ which was defined in Eq. (3.43) can be constructed as:

$$T_m(r,t) = \underset{1\times 2}{\underset{\sim}{\phi}_m(r)} \; \underset{2\times 1}{\underset{\sim}{T}_m(t)} = \left[(1 - \frac{r}{L}) \; (\frac{r}{L})\right] \begin{Bmatrix} T_i \\ T_j \end{Bmatrix} \qquad (3.46)$$

where

T_i, T_j = temperatures of nodes i and j.

r = local coordinate measured along surface of element from node i to node j.

L = length of boundary surface: $\left[(x_i - x_j)^2 + (y_i - y_j)^2\right]^{1/2}$

$\left.\begin{matrix} x_i, x_j \\ y_i, y_j \end{matrix}\right\}$ = global coordinates of nodes i and j.

Assuming the planar boundary element to have unit width, the heat flux vector $\underset{\sim}{q}_m^2$ from Eq. (3.44) can now be determined:

$$\underset{2\times 1}{\underset{\sim}{q}_m^2} = \int_0^1 \int_0^L h_m^c \, T_e \begin{bmatrix} 1 - \frac{r}{L} \\ \frac{r}{L} \end{bmatrix} dr\, dt$$

and solving we have:

$$\begin{aligned} (q_i^2)_m &= \frac{h_m^c L}{2} T_e \\ (q_j^2)_m &= \frac{h_m^c L}{2} T_e \end{aligned} \qquad (3.47)$$

The element conductivity matrix $\underset{\sim}{K}_m^2$ for a planar boundary surface m can be similarly found from Eq. (3.45) as:

$$\underset{\sim}{K}_m^2 = \int_0^1\int_0^L h_m^c \begin{bmatrix} 1 - \frac{r}{L} \\ \frac{r}{L} \end{bmatrix} \left[(1 - \frac{r}{L}) \ (\frac{r}{L})\right] dr\ dt$$

and solving we have:

$$(K_{ii}^2)_m = \frac{h_m^c L}{3}$$

$$(K_{jj}^2)_m = \frac{h_m^c L}{3} \qquad (3.48)$$

$$(K_{ij}^2)_m = \frac{h_m^c L}{6}$$

(b) Axisymmetric Solid

Consider the axisymmetric convection boundary element in Fig. 3.3(b) which lies between nodes i and j. The temperature interpolation vector $\underset{\sim}{\phi}_m$ which describes the variation of temperature along this element is identical to the one constructed for the planar solid in Eq. (3.46).

The differential surface area dS_m for a segment of angle θ can be computed as:

$$dS_m = dr \cdot X_1(r)\ d\theta \qquad (3.49)$$

where

r = local coordinate for boundary element surface measured from node i to node j.

$X_1(r)$ = radius to any point r between nodes i and j.

From Eq. (3.1) we have:

$$X_1(r) = \underset{\sim}{\phi}_m \underset{\sim}{x}_m = \left[(1 - \frac{r}{L}) \; (\frac{r}{L})\right] \begin{Bmatrix} x_i \\ x_j \end{Bmatrix} \tag{3.50}$$

Substituting Eq. (3.50) into (3.49) and assuming a unit radian segment ($d\theta = 1$), the heat flux vector $\underset{\sim}{q}_m^2$ for the axisymmetric boundary element m may now be found from Eq. (3.44):

$$\underset{2\times 1}{\underset{\sim}{q}_m^2} = \int_0^1\int_0^L h_m^c \, T_e \begin{bmatrix} 1 - \frac{r}{L} \\ \frac{r}{L} \end{bmatrix} \left[(1 - \frac{r}{L}) \; (\frac{r}{L})\right] \begin{Bmatrix} x_i \\ x_j \end{Bmatrix} dr \, d\theta$$

and solving we have:

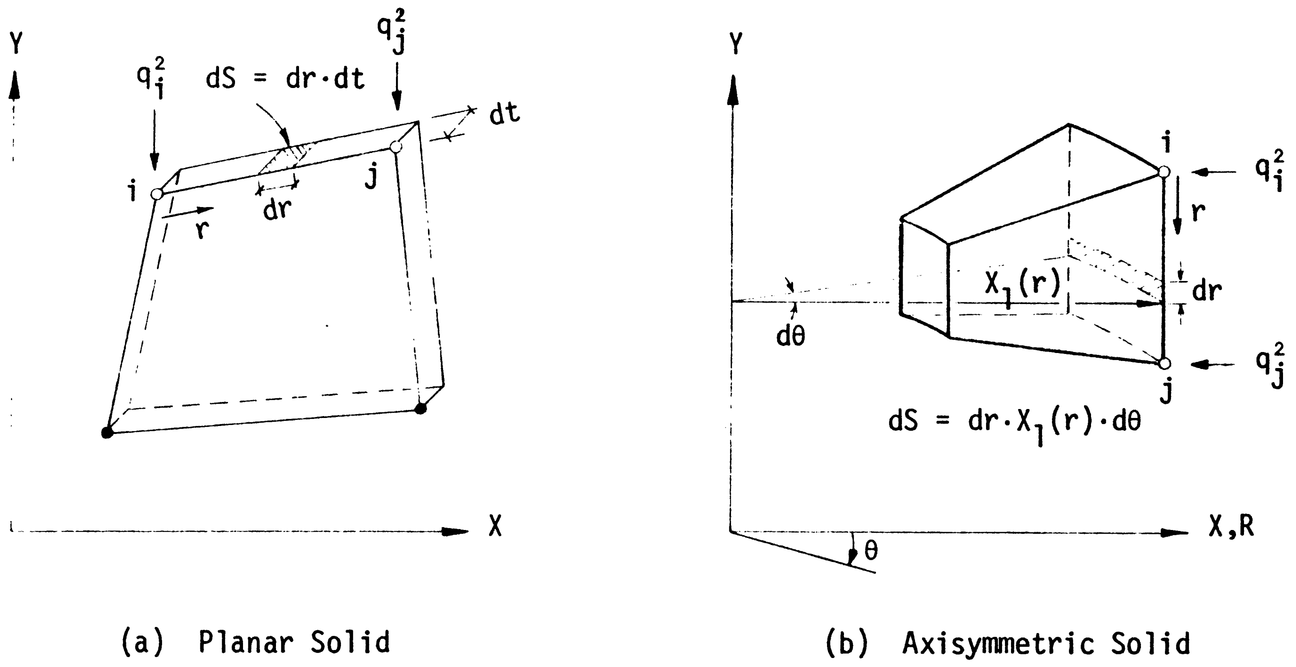
$$\begin{aligned} (q_i^2)_m &= \frac{h_m^c L}{6} T_e \, [2x_i + x_j] \\ (q_j^2)_m &= \frac{h_m^c L}{6} T_e \, [x_i + 2x_j] \end{aligned} \tag{3.51}$$

Similarly, the element conductivity matrix $\underset{\sim}{K}_m^2$ for an axisymmetric boundary element may be determined from Eq. (3.45):

$$\underset{2\times 2}{\underset{\sim}{K}_m^2} = \int_0^1\int_0^L h_m^c \begin{bmatrix} 1 - \frac{r}{L} \\ \frac{r}{L} \end{bmatrix} \left[(1 - \frac{r}{L}) \; (\frac{r}{L})\right] \left[(1 - \frac{r}{L}) \; (\frac{r}{L})\right] \begin{Bmatrix} x_i \\ x_j \end{Bmatrix} dr \, d\theta$$

and solving we have:

$$\begin{aligned} (K_{ii}^2)_m &= \frac{h_m^c L}{4} [x_i + \frac{1}{3} x_j] \\ (K_{jj}^2)_m &= \frac{h_m^c L}{4} [\frac{1}{3} x_i + x_j] \\ (K_{ij}^2)_m &= \frac{h_m^c L}{12} [x_i + x_j] \end{aligned} \tag{3.52}$$



(a) Planar Solid

(b) Axisymmetric Solid

FIG. 3.3 CONVECTION BOUNDARY ELEMENTS FOR PLANAR AND AXISYMMETRIC SOLIDS

C. Radiation Heat Transfer

The rate of heat transfer by radiation between two surfaces can be found by considering both surfaces to be gray and idealizing them as two infinitely large parallel plates. The net rate of radiative heat transfer per unit area between the surfaces can then be expressed as [29]:

$$q^3 = V\sigma \left[\frac{1}{\frac{1}{\varepsilon_r} + \frac{1}{\varepsilon_s} - 1} \right] [\bar{T}_r^4 - \bar{T}_s^4] = h^r \cdot [\bar{T}_r^4 - \bar{T}_s^4] \qquad (3.53)$$

where V = radiation view factor.

σ = Stefan-Boltzmann constant.

ε_r = emissivity of the radiation source.

ε_s = emissivity of the boundary surface.

$\bar{T}_r$ = absolute temperature of the radiation source.

$\bar{T}_s$ = absolute temperature of the boundary surface.

If we consider a radiation boundary element m of unit thickness which lies between nodes i and j, the rate at which heat is transferred to the nodes will be approximately:

$$\begin{aligned} (Q_i^3)_m &= \frac{1}{2} h_m^r L_m [(\bar{T}_r^4)_m - (\bar{T}_s^4)_m] \\ (Q_j^3)_m &= \frac{1}{2} h_m^r L_m [(\bar{T}_r^4)_m - (\bar{T}_s^4)_m] \end{aligned} \qquad (3.54)$$

where h_m^r = radiation heat transfer coefficient defined in Eq. (3.53).

L_m = length of boundary surface between nodes i and j: $\left[(x_i^2 - x_j^2) + (y_i^2 - y_j^2)\right]^{1/2}$

$\left.\begin{matrix} x_i, x_j \\ y_i, y_j \end{matrix}\right\}$ = global coordinates of nodes i and j.

$(\bar{T}_r)_m$ = absolute temperature of radiation source for boundary element m.

$(\bar{T}_s)_m$ = average value of absolute surface temperature: $\left(\frac{1}{2}[\bar{T}_i + \bar{T}_j]\right)$

The temperature of the radiation source $\bar{T}_r$ must be given as a prescribed function of time, and the value of the average absolute surface temperature $\bar{T}_s$ will depend on the unknown nodal temperatures $\bar{T}_i$ and $\bar{T}_j$. For the finite element program DOT which is described in Appendix A, the value of $\bar{T}_s$ was determined at the midpoint of a particular time step by assuming a linear variation of nodal temperatures over the previous time step.

D. Internal Heat Generation

The rate of internal heat generation p_m within any element m is assumed to initiate at some starting time t_o and thereafter can vary with time. Thus

$$\begin{aligned} p_m &= 0 \quad ; \quad t < t_o \\ p_m &= f(t) \quad ; \quad t > t_o \end{aligned} \qquad (3.55)$$

where f(t) is a prescribed function of time.

The vector of nodal fluxes for element m at any time $t > t_o$ can then be computed as:

$$q_m^4 = \int_{V_m} \rho_m \, p_m \, \underline{b}_m^T \, dV_m \qquad (3.56)$$

where

ρ_m = mass density.

p_m = rate of internal heat generation at time t.

$\underline{b}_m$ = temperature interpolation vector (Eq. (3.5)).

V_m = volume of element.

The integration in Eq. (3.56) is evaluated numerically using Gaussian quadrature and the heat flux components in $\underline{q}_m^4$ are assembled directly into the heat flux vector $\underline{Q}^4(t)$.

The basic equations for the formulation of the finite element equilibrium equation (3.20) are summarized in Table 3.1.

TABLE 3.1

MATRIX EQUILIBRIUM EQUATIONS FOR HEAT TRANSFER ANALYSIS

$$\underline{C}(T)\,\dot{\underline{T}}(t) + \underline{K}(T)\,\underline{T}(t) = \underline{Q}(t) \qquad (3.20)$$

where

$$\underline{C}(T) = \sum_{m=1}^{M} \underline{C}_m(T) \qquad (3.32)$$

$$\underline{K}(T) = \sum_{m=1}^{M} \underline{K}_m(T) \qquad (3.21)$$

$$\underline{Q}(t) = \sum_{m=1}^{M} \underline{Q}_m(t) \qquad (3.41)$$

and

$$\underline{C}_m(T) = \int_{V_m} \rho_m\, c_m(T)\, \underline{b}_m^T\, \underline{b}_m\, dV_m \qquad (3.33)$$

$$\underline{K}_m(T) = \int_{V_m} \underline{a}_m^T\, \underline{k}_m(T)\, \underline{a}_m\, dV_m \qquad (3.22)$$

$$+ \int_{S_m} h_m^c(T)\, \underline{\phi}_m^T\, \underline{\phi}_m\, dS_m \qquad (3.45)$$

$$\underline{Q}_m(t) = \underline{Q}_m^1(t) + \int_{S_m} h_m^c\, T_e(t)\, \underline{\phi}_m^T\, dS_m + \underline{Q}_m^3(t) + \int_{V_m} \rho_m\, p_m(t)\, \underline{b}_m^T\, dV_m \qquad \left\{\begin{matrix}(3.41)\\(3.44)\\(3.54)\\(3.56)\end{matrix}\right.$$

4. BOUNDARY CONDITIONS

4.1 NODAL TEMPERATURE AND HEAT FLUX BOUNDARY CONDITIONS

If a surface element between nodes i and j is exposed to a known heat flux, the heat flow can be lumped at nodes i and j by using a method based on the tributary surface area. These lumped heat flow quantities can then be directly assembled into the system heat flux vector $\underline{Q}^1(t)$. For nodes on insulated surfaces, the external heat flow is zero.

At certain points in the system, the temperatures may be specified as a function of time. These temperature boundary conditions are similar to the displacement boundary conditions in structural analysis. A commonly used procedure which accounts for these boundary conditions is to partition the matrix equilibrium equation (3.20) in terms of the unknown and specified temperatures [22].

An alternative and very effective computational approach for treating specified nodal temperatures is outlined as follows. Consider node i in the finite element system shown in Fig. 4.1 where the temperature has a known value $T_i(t)$. Assume that an imaginary element which has a large volume and a high thermal conductivity is connected to node i only. Then the diagonal coefficient term in the system conductivity matrix $\underline{K}$ for nodal point i is:

$$K_{ii} = K'_{ii} + k \qquad (4.1)$$

where $k \gg K'_{ii}$. For the DOT and DETECT computer programs described in Appendices A and B, the value of k used is 10^{10}.

In a structural analysis problem, this approach is equivalent to adding an imaginary spring to degree of freedom i which has a very large stiffness k_s and then specifying a load equal to $k_s u_i$ at this degree of freedom which produces the specified displacement u_i.

Note that the system conductivity matrix $\underset{\sim}{K}$ does not become ill-conditioned by adding a large number k to the existing diagonal coefficient K'_{ii}. Only the addition of large numbers to the off-diagonal terms can result in numerical instability of the $\underset{\sim}{K}$ matrix.

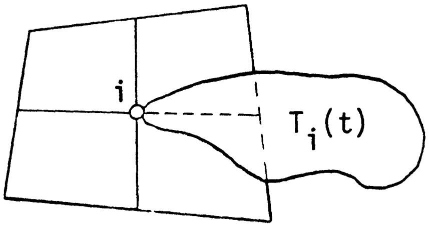


FIG. 4.1 TEMPERATURE BOUNDARY CONDITION IDEALIZATION

4.2 CONVECTION BOUNDARY CONDITION

For linear free convection, the rate of heat flow across a given boundary layer is proportional to the difference in temperature between the surface of the solid T_s and the external environment T_e. The necessary modifications to the equilibrium as a result of surface heat transfer have been previously described in Eqs. (3.44) and (3.45). It is of interest to note that the surface temperature of a solid subjected to convection heat transfer closely follows the temperature variation of the external environment. The amount by which the surface temperature lags behind the environmental temperature is controlled by the magnitude of the surface heat transfer coefficient h^c in Eq. (3.42). By using a sufficiently large h^c ($\approx 10^{10}$), the surface temperature will follow the environmental temperature variation exactly.

4.3 RADIATION BOUNDARY CONDITION

The net rate of radiant heat transfer between a blackbody surface and an enclosure which completely surrounds it whose surface is also black, i.e. absorbs all the radiant energy incident upon it, is given by:

$$q_r = \sigma A_s [\bar{T}_r^4 - \bar{T}_s^4] \qquad (4.2)$$

The heat flow rate q_r will be in BTU/hr if the surface area A_s is in sq. ft., the surface and radiation source temperatures $\bar{T}_s$ and $\bar{T}_r$ are in degrees Rankine (^oR) and the Stefan-Boltzmann constant σ has the value of 0.1714×10^{-8} BTU/hr-ft^2-^oR.

Most surfaces encountered in engineering applications do not behave like blackbodies. If neither of the two bodies is a perfect radiator, i.e. if they emit radiation at a lower rate than blackbodies, they are called "gray" bodies.

For the DOT program described in Appendix A, the rate of radiation heat transfer between two gray surfaces is found by considering both the surface and the radiation source to be two infinitely large parallel plates. The required modifications to the system heat flux vector $\underline{Q}^3(t)$ as a result of radiation heat transfer have been described previously in Eq. (3.54).

4.4 COOLING PIPES

In the construction of mass concrete structures, the heat given off by the hydrating cement becomes entrapped in the mass under nearly adiabatic conditions and results in the concrete temperature increasing rapidly above the placement temperature. The subsequent decrease in temperature to a level corresponding to the ambient air temperature causes the concrete to contract and may result in random structural cracking if the magnitude of the temperature drop is not controlled. This temperature drop can be minimized by placing the concrete at a low initial temperature and by artificially cooling the concrete by means of embedded pipes.

The previously developed matrix equation governing heat flow equilibrium (Eq. (3.20)) must be modified in order to account for the heat transferred out of the solid by a cooling pipe. Consider a cooling pipe located at node i in a finite element system. Assuming that the cooling pipe is thin-walled, the temperature on the outer surface of the pipe is approximately equal to the temperature of the water T_w in the pipe. Having made this assumption, the exact temperature distribution between a cooling pipe located at node i of element m and an adjacent nodal point a distance "a" away from the pipe is shown in Fig. 4.1.

The rate at which heat flows out of the solid and into the cooling pipe is given by [22]:

$$q = -kA \cdot \frac{\partial T}{\partial r} = -k \cdot 2\pi rt \cdot \frac{\partial T}{\partial r} \qquad (4.3)$$

where k = average thermal conductivity of the finite elements surrounding the cooling pipe.

r = distance from the center of the pipe.

T = temperature field.

t = thickness of element.

By separating the variables in Eq. (4.3) and integrating between T_w at R and T_a at "a" we find:

$$q = \frac{2\pi kt}{\ln\left[\frac{a}{R}\right]}(T_w - T_a) \qquad (4.4)$$

where T_a = temperature at the node point a distance "a" from the pipe.

T_w = temperature of the cooling water.

R = radius of the cooling pipe.

In the finite element solution, the temperature distribution within the elements adjacent to the cooling pipe is assumed to be linear. If we associate an effective tributary area having a radius of a/2 with each cooling pipe in the system, then the rate at which heat flows into the pipe from this area can be found from Eq. (4.3) as:

$$q = -2\pi kt\left[\frac{a}{2}\right]\left(\frac{T_a - T_i}{a}\right) = \pi kt\cdot(T_i - T_a) \qquad (4.5)$$

where T_i = apparent temperature at node i in the finite element system.

Combining Eqs. (4.4) and (4.5) and eliminating T_a we can obtain the rate at which heat is removed from the system by a cooling pipe:

$$q = H \cdot (T_w - T_i) \tag{4.6}$$

where $$H = \frac{2\pi kt}{\ln\left[\frac{a}{R}\right] - 2}$$

It is of interest to note that for an element size "a" equal to 7.4 times the radius R of the pipe, $\ln\left[\frac{a}{R}\right] = 2$ and the value of H is infinite. For this element size, therefore, the apparent nodal temperature T_i is equal to the cooling water temperature T_w. For other mesh spacings the correct value of H can be computed from Eq. (4.6). In case the elements surrounding the cooling pipe are not all the same size, a weighted value of "a" should be used in Eq. (4.6) representing the average distance to the adjacent nodes.

The cooling pipe boundary condition equation (4.6) is recognized to be of the same form as the convection boundary condition equation (3.42). Similar modifications, therefore, can be made to the system conductivity matrix $\underline{K}$ and heat flux vector $\underline{Q}$ as were made in Eqs. (3.47) and (3.48).

For a cooling pipe located at a typical nodal point i in a finite element system, the following modifications must be made to the system conductivity and heat flux matrices:

$$\begin{aligned} K_{ii} &= K'_{ii} + H \\ Q_i &= Q'_i + HT_w \end{aligned} \tag{4.7}$$

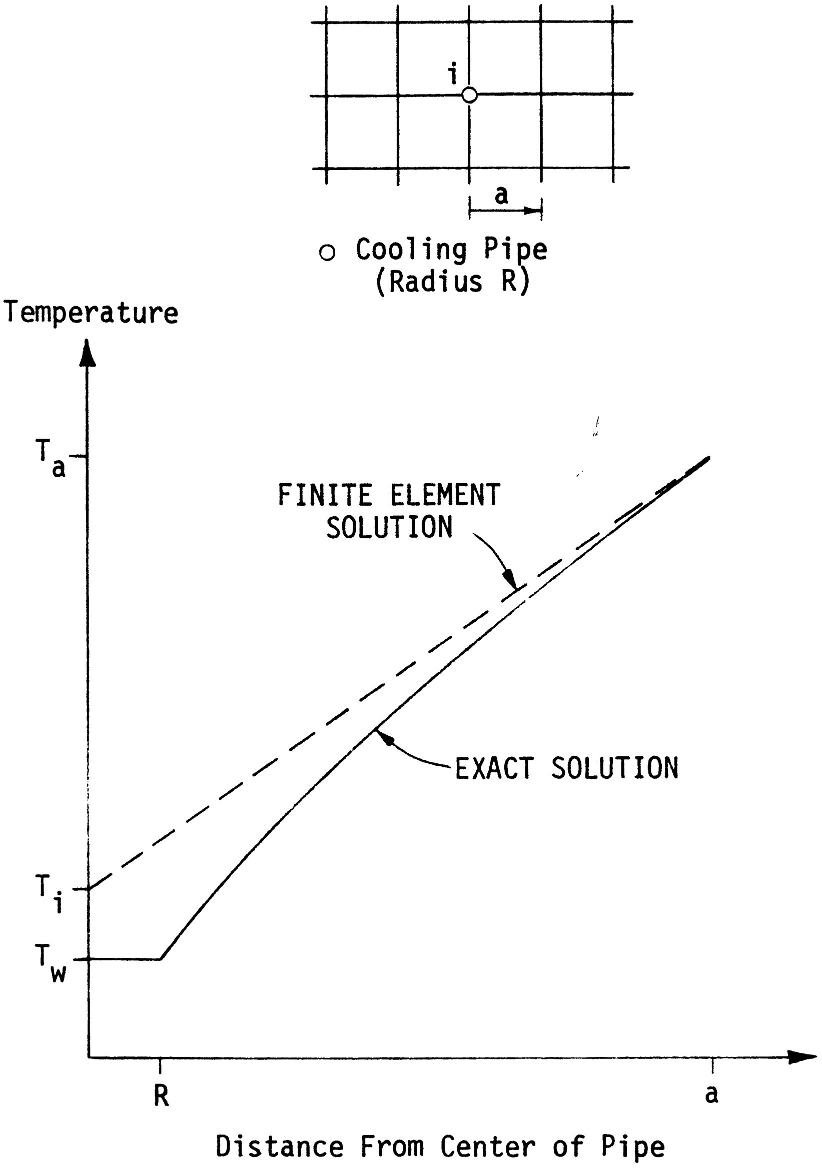


FIG. 4.1 TEMPERATURE DISTRIBUTION NEAR A COOLING PIPE

5. NUMERICAL SOLUTION OF THE HEAT FLOW EQUILIBRIUM EQUATION

The heat flow equilibrium equation (3.20) represents a set of first order nonlinear differential equations. Numerical integration schemes for solving this equation will be discussed for both linear systems having constant thermal properties and nonlinear systems in which the thermal properties are temperature dependent.

5.1 LINEAR SYSTEMS

At any instant "t" in time, the linear heat conduction equilibrium equation for a finite element system given by Eq. (3.20) can be expressed as:

$$\underline{C}\,\dot{\underline{T}}_t + \underline{K}\,\underline{T}_t = \underline{Q}_t \qquad (5.1)$$

Similarly, at time t + Δt we have:

$$\underline{C}_{t+\Delta t}\,\dot{\underline{T}}_{t+\Delta t} + \underline{K}_{t+\Delta t}\,\underline{T}_{t+\Delta t} = \underline{Q}_{t+\Delta t} \qquad (5.2)$$

The solution to this set of first order differential equations can be obtained by using a step-by-step integration (or time marching) procedure which assumes a linear variation of temperature over the time step Δt (Fig. 5.1). This is a simple method which has proven to be both efficient and accurate [3,24].

Using the assumption of linear temperature variation over the time step Δt, the time rate-of-change in temperature $\dot{\underline{T}}$ at any time ξ in the interval {t,t+Δt} is:

$$\dot{\underline{T}}(\xi) = \frac{1}{\Delta t}\,(\underline{T}_{t+\Delta t} - \underline{T}_t) \qquad (5.3)$$

Hence, at time t = t + Δt we have:

$$\dot{\underline{T}}_{t+\Delta t} = \frac{1}{\Delta t}\,(\underline{T}_{t+\Delta t} - \underline{T}_t) \qquad (5.4)$$

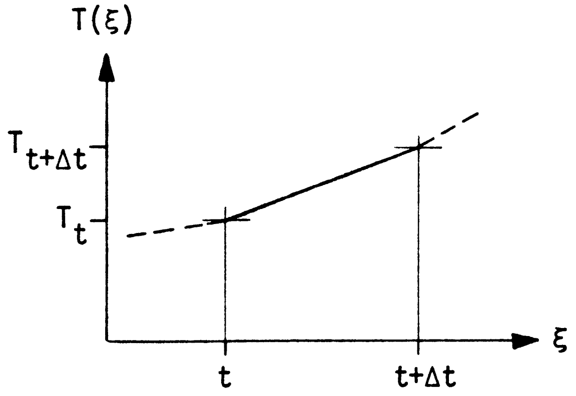


FIG. 5.1 TIME VARIATION OF NODAL TEMPERATURES T OVER A SHORT TIME PERIOD Δt

The matrix equation for heat flow equilibrium at time t+Δt can now be found by substituting Eq. (5.4) into Eq. (5.2). The resulting equation is:

$$\left[\frac{1}{\Delta t}\underline{C} + \underline{K}\right]\underline{T}_{t+\Delta t} = \underline{Q}_{t+\Delta t} + \frac{1}{\Delta t}\underline{C}\,\underline{T}_t \qquad (5.5)$$

Defining,

$$\underline{K}^* = \frac{1}{\Delta t}\underline{C} + \underline{K} \qquad (5.6)$$

$$\underline{Q}^*_{t+\Delta t} = \underline{Q}_{t+\Delta t} + \frac{1}{\Delta t}\underline{C}\,\underline{T}_t \qquad (5.7)$$

Then Eq. (5.5) simplifies to:

$$\underline{K}^*\,\underline{T}_{t+\Delta t} = \underline{Q}^*{}_{t+\Delta t} \qquad (5.8)$$

Since the matrices $\underline{C}$, $\underline{K}$, $\underline{Q}_{t+\Delta t}$ and $\underline{T}_t$ are known at the start of the time interval {t,t+Δt}, Eq. (5.8) can therefore be solved directly for the nodal temperatures $\underline{T}_{t+\Delta t}$ at the end of the time interval. The step-by-step solution procedure for solving Eq. (5.8) is summarized in Table 5.1.

Since the "effective" conductivity matrix $\underline{K}^*$ is not a function of time, it can be formed once for a given geometry and triangularized. In each subsequent time step, therefore, it is only necessary to define the thermal load vector $\underline{Q}_{t+\Delta t}$ and then solve by simple back-substitution for the system temperatures $\underline{T}_{t+\Delta t}$. The matrix $\underline{K}^*$ is normally banded and its triangularized form is banded as well, so a large amount of computer storage is usually not required for placing $\underline{K}^*$ in core.

It should be noted that at time zero, an initial distribution of nodal temperatures $\underline{T}_o$ may exist. For this case, the "effective" heat flux vector $\underline{Q}^*_{\Delta t}$ at the end of the first time interval is given by:

$$\begin{aligned} \underline{Q}^*_{\Delta t} &= \underline{Q}_{\Delta t} + \frac{1}{\Delta t}\underline{C}\,\underline{T}_o \\ &= \underline{Q}_{\Delta t} + \underline{E}_o \end{aligned} \tag{5.9}$$

where $\underline{E}_o$ is called the initial resistance vector. In general:

$$\underline{Q}^*_{t+\Delta t} = \underline{Q}_{t+\Delta t} + \underline{E}_t \tag{5.10}$$

where

$$\underline{E}_t = \frac{1}{\Delta t}\underline{C}\,\underline{T}_t$$

TABLE 5.1

SUMMARY OF THE STEP-BY-STEP SOLUTION METHOD FOR LINEAR HEAT TRANSFER

INITIAL CALCULATIONS

1. Form $\underline{K}$ and $\underline{C}$; initialize $\underline{T}_o$.

2. Modify $\underline{K}$ for convection and cooling pipe boundary conditions.

3. Modify $\underline{K}$ for temperature boundary conditions.

4. Compute the effective system matrices:

$$\underline{C}^* = \frac{1}{\Delta t}\underline{C}$$

$$\underline{K}^* = \underline{K} + \underline{C}^*$$

5. Compute the initial resistance vector:

$$\underline{E}_o = \underline{C}^* \underline{T}_o$$

6. Triangularize $\underline{K}^*$.

FOR EACH SOLUTION TIME STEP

1. Define the heat flow vector $\underline{Q}_{t+\Delta t}$ at time t+Δt.

2. Compute the effective heat flow vector:

$$\underline{Q}^*_{t+\Delta t} = \underline{Q}_{t+\Delta t} + \underline{E}_t$$

3. Solve the effective heat flow equilibrium equation for the nodal temperatures $\underline{T}_{t+\Delta t}$ at time t+Δt.

$$\underline{K}^* \underline{T}_{t+\Delta t} = \underline{Q}^*{}_{t+\Delta t}$$

4. Compute the resistance vector for the next time step:

$$\underline{E}_{t+\Delta t} = \underline{C}^* \underline{T}_{t+\Delta t}$$

5. Repeat for the next time step.

5.2 STABILITY OF THE NUMERICAL INTEGRATION SCHEME

The step-by-step integration procedure described in the previous section assumed a linear variation of temperature over the time step Δt. This algorithm is unconditionally stable for a heat conduction problem if the solution of the initial value problem is bounded (i.e. no error growth) for any time step Δt. This is to say that any round-off error in the predicted temperatures does not grow with time as the solution proceeds.

5.2.1 Modal Heat Flow Equations

Consider the heat flow equilibrium equation:

$$\underline{C}\,\dot{\underline{T}}(t) + \underline{K}\,\underline{T}(t) = \underline{Q}(t) \tag{5.11}$$

For a solution to the homogeneous part of this equation, assume:

$$\underline{T}(t) = e^{-\lambda t}\,\underline{\phi} \tag{5.12}$$

Thus,

$$\underline{C}\,[-\lambda e^{-\lambda t}\,\underline{\phi}] + \underline{K}\,[e^{-\lambda t}\,\underline{\phi}] = \underline{0}$$

Because the value of $e^{-\lambda t}$ can never equal zero, we therefore obtain a statement of the generalized eigenvalue problem:

$$\underline{K}\,\underline{\phi} = \lambda\,\underline{C}\,\underline{\phi} \tag{5.13}$$

Equation (5.13) is an nth order equation where n is the number of temperature degrees of freedom in the system. The n solutions to Eq. (5.13) can be stated as:

$$\underline{K}\,\underline{\Phi} = \underline{C}\,\underline{\Phi}\,\underline{\Lambda} \tag{5.14}$$

where the columns in $\underline{\Phi}$ are the $\underline{C}$-orthonormalized (i.e. $\underline{\Phi}^T\underline{C}\,\underline{\Phi} = \underline{I}$) eigenvectors (thermal modes) $\underline{\phi}_1 \ldots . \underline{\phi}_n$, and $\underline{\Lambda}$ is a diagonal matrix containing the eigenvalues (thermal frequencies) $\lambda_1 \ldots . \lambda_n$.

By using the transformation $\underline{T} = \underline{\Phi}\,\underline{X}$ (and $\dot{\underline{T}} = \underline{\Phi}\,\dot{\underline{X}}$), the heat flow equilibrium Eq. (5.11) can be written in the eigenvector basis as:

$$\underline{C}\,\underline{\Phi}\,\dot{\underline{X}} + \underline{K}\,\underline{\Phi}\,\underline{X} = \underline{Q} \tag{5.15}$$

Substituting Eq. (5.14) into (5.15):

$$\underline{C}\,\underline{\Phi}\,\dot{\underline{X}} + \underline{C}\,\underline{\Phi}\,\underline{\Lambda}\,\underline{X} = \underline{Q} \tag{5.16}$$

Pre-multiplying Eq. (5.16) by $\underline{\Phi}^T$:

$$\underline{\Phi}^T\,\underline{C}\,\underline{\Phi}\,\dot{\underline{X}} + \underline{\Phi}^T\,\underline{C}\,\underline{\Phi}\,\underline{\Lambda}\,\underline{X} = \underline{\Phi}^T\,\underline{Q} \tag{5.17}$$

Because the eigenvectors $\underline{\Phi}$ are $\underline{C}$-orthonormalized, Eq. (5.17) reduces to the form:

$$\dot{\underline{X}} + \underline{\Lambda}\,\underline{X} = \underline{\Phi}^T\,\underline{Q} \tag{5.18}$$

These equations are uncoupled and can readily be solved by a number of numerical procedures.

To study the stability and accuracy of the iteration scheme, it is noted that the integration of the heat flow equilibrium equation (5.11) in the original finite element coordinate basis is equivalent to the integration of the n uncoupled first order differential equations (5.18) in the thermal mode basis. Therefore, for the purpose of a stability analysis, we only need to study the integration algorithm as applied to a typical equation "i" in Eq. (5.18):

$$\dot{x} + \lambda_i x = q \tag{5.19}$$

where

$$q = \underline{\Phi}_i^T\,\underline{Q}$$

5.2.2 Step-By-Step Integration Procedure

In order to express Eq. (5.19) in a difference form, let the temperature x_t and the time rate-of-change of temperature $\dot{x}_t$ be known at time t. By assuming that the temperature varies linearly in the time interval $\{t,t+\Delta t\}$, the temperature gradient $\dot{x}_{t+\Delta t}$ at time $t+\Delta t$ is:

$$\dot{x}_{t+\Delta t} = \frac{1}{\Delta t}(x_{t+\Delta t} - x_t) \tag{5.20}$$

Substituting Eq. (5.20) into (5.19), the resulting heat flow equilibrium equation to be satisfied in the ith mode at time $t+\Delta t$ is:

$$\frac{1}{\Delta t}(x_{t+\Delta t} - x_t) + \lambda_i x_{t+\Delta t} = q_{t+\Delta t}$$

or

$$x_{t+\Delta t} = Ax_t + Lq_{t+\Delta t} \tag{5.21}$$

where

$$A = \frac{1}{1 + \lambda_i \Delta t} \tag{5.22}$$

$$L = \frac{\Delta t}{1 + \lambda_i \Delta t} \tag{5.23}$$

The quantity A is the difference approximation operator and L is the heat flux operator as applied to the ith uncoupled modal heat flow equation.

The recursion relation in Eq. (5.21) can be used to study the stability and accuracy of the integration scheme. In general, the solution at any time $t + m\Delta t$ (where m is a positive integer) is given by:

$$x_{t+m\Delta t} = A^m x_t + A^{m-1}Lq_{t+\Delta t} + \dots + Lq_{t+m\Delta t} \tag{5.24}$$

5.2.3 Stability

Consider Eq. (5.24) with q equal to zero:

$$x_{t+m\Delta t} = A^m x_t \tag{5.25}$$

The integration method is unconditionally stable if A^m is bounded for large values of m (i.e. as $m \to \infty$) and for any time step Δt. This stability condition may be stated as:

$$A \leq 1 \tag{5.26}$$

It is noted that if $A < 1$, then $A^m \to 0$ as $m \to \infty$, and the smaller the value of A the more rapid the convergence. Recalling Eq. (5.22), the stability condition requires that:

$$A = \frac{1}{1 + \lambda_i \Delta t} \leq 1$$

or

$$1 + \lambda_i \Delta t \geq 1 \tag{5.27}$$

Since the thermal frequency λ_i for the ith mode must satisfy the relation $\lambda_i \geq 0$, the integration algorithm is unconditionally stable for all positive values of the selected time step ($\Delta t > 0$). The accuracy of the integration scheme clearly depends on the magnitude of $\lambda_i \Delta t$.

5.3 NONLINEAR SYSTEMS

In the analysis of nonlinear systems, direct integration must be used. At time t + Δt, the exact equation of heat flow equilibrium is:

$$\underset{\sim}{C}_{t+\Delta t}\,\dot{\underset{\sim}{T}}_{t+\Delta t} + \underset{\sim}{K}_{t+\Delta t}\,\underset{\sim}{T}_{t+\Delta t} = \underset{\sim}{Q}_{t+\Delta t} \tag{5.28}$$

where

$\underset{\sim}{C}_{t+\Delta t}$ = system heat capacity matrix at time t + Δt.

$\underset{\sim}{K}_{t+\Delta t}$ = system thermal conductivity matrix at t + Δt.

$\underset{\sim}{T}_{t+\Delta t}$ = vector of nodal point temperatures at t + Δt.

$\dot{\underset{\sim}{T}}_{t+\Delta t}$ = vector of the time rate-of-change of the nodal point temperatures at time t + Δt.

$\underset{\sim}{Q}_{t+\Delta t}$ = vector of the externally supplied nodal heat fluxes at time t + Δt.

In order to solve Eq. (5.28), we make the approximation that:

$$\begin{aligned}\underset{\sim}{C}_{t+\Delta t} &\cong \underset{\sim}{C}_{t}\\ \underset{\sim}{K}_{t+\Delta t} &\cong \underset{\sim}{K}_{t}\end{aligned} \tag{5.29}$$

Equation (5.28) now becomes:

$$\underset{\sim}{C}_{t}\,\dot{\underset{\sim}{T}}_{t+\Delta t} + \underset{\sim}{K}_{t}\,\underset{\sim}{T}_{t+\Delta t} \doteq \underset{\sim}{Q}_{t+\Delta t} \tag{5.30}$$

Using the assumption in Eq. (5.4) that the temperature variation over the time step Δt is linear, we can substitute Eq. (5.4) into Eq. (5.30) to obtain the following approximation:

$$\underset{\sim}{C}_{t}\,\frac{1}{\Delta t}\left(\underset{\sim}{T}_{t+\Delta t} - \underset{\sim}{T}_{t}\right) + \underset{\sim}{K}_{t}\,\underset{\sim}{T}_{t+\Delta t} \doteq \underset{\sim}{Q}_{t+\Delta t} \tag{5.31}$$

Subtracting $\underline{K}_t\underline{T}_t$ from both sides of Eq. (5.31) we have:

$$\frac{1}{\Delta t}\underline{C}_t\,\Delta\underline{T}_t + \underline{K}_t\,\Delta\underline{T}_t \doteq \underline{Q}_{t+\Delta t} - \underline{K}_t\underline{T}_t \qquad (5.32)$$

where

$$\Delta\underline{T}_t = \underline{T}_{t+\Delta t} - \underline{T}_t \qquad (5.33)$$

Equation (5.32) can be rewritten as:

$$\left[\frac{1}{\Delta t}\underline{C}_t + \underline{K}_t\right]\Delta\underline{T}_t \doteq \underline{Q}_{t+\Delta t} - \underline{K}_t\underline{T}_t$$

or

$$\underline{K}_t^{*}\,\Delta\underline{T}_t \doteq \underline{Q}_{t+\Delta t} - \underline{F}_t^{k} \qquad (5.34)$$

where

$$\underline{K}_t^{*} = \frac{1}{\Delta t}\underline{C}_t + \underline{K}_t$$

$$\underline{F}_t^{k} = \underline{K}_t\underline{T}_t$$

The incremental equation for heat flow equilibrium given in Eq. (5.34) is equivalent to applying Eq. (5.8) in which $\underline{K}$ is updated in each time step. The solution for the nodal temperatures at time t + Δt is given by:

$$\underline{T}_{t+\Delta t} = \underline{T}_t + \Delta\underline{T}_t \qquad (5.35)$$

Because of errors introduced by the linearization of Eq. (5.31), the temperatures $T_{t+\Delta t}$ resulting from Eqs. (5.34) and (5.35) are only an approximation to the actual nodal temperatures at time t + Δt. These errors may be corrected by using an iterative technique for heat flow equilibrium. An efficient step-by-step solution procedure for nonlinear heat conduction problems is summarized in Table 5.2 which includes the option of heat flow equilibrium iteration.

TABLE 5.2

SUMMARY OF THE STEP-BY-STEP SOLUTION METHOD FOR NONLINEAR HEAT TRANSFER

INITIAL CALCULATIONS

1. Define all input, including initial temperatures $\underline{T}_o$.

2. Form $\underline{K}$ and $\underline{C}$ based on properties at $\underline{T}_o$.

3. Modify $\underline{K}$ for convection and cooling pipe boundary conditions.

4. Modify $\underline{K}$ for temperature boundary conditions.

5. Compute the internal heat flow due to conduction at time zero:

$$\underline{F}_o^k = \underline{K}_o \underline{T}_o$$

6. Compute the effective system matrices:

$$\underline{C}^* = \frac{1}{\Delta t} \underline{C}$$

$$\underline{K}^* = \underline{K} + \underline{C}^*$$

7. Triangularize $\underline{K}^*$.

8. Initialize:

   a. Time step counter: $i = 0$, $t = t_{START}$

   b. Convergence tolerance: TOL

   c. Maximum number of iterations: NITER

FOR EACH TIME STEP

1. Increment time step counter: $i = i + 1$ and $t = t + \Delta t$

2. Test for equation reformation:

   NO : Advance to Step 9

   YES : Continue

3. Form new system matrices $\underline{K}_t$ and $\underline{C}_t$ based on properties $\underline{T}_t$.

4. Modify $\underline{K}_t$ for convection and cooling pipe boundaries.

5. Modify $\underline{K}_t$ for temperature boundary conditions.

TABLE 5.2 (CON'T)

6. Compute internal heat flow due to conduction:

$$\underline{F}_t^k = \underline{K}_t \underline{T}_t$$

7. Compute the new effective system matrices:

$$\underline{C}_t^* = \frac{1}{\Delta t} \underline{C}_t$$

$$\underline{K}_t^* = \underline{K}_t + \underline{C}_t^*$$

8. Triangularize $\underline{K}_t^*$.

9. Define the heat flow vector $\underline{Q}_{t+\Delta t}$ at time t + Δt.

10. Compute the effective heat flow vector:

$$\underline{Q}_{t+\Delta t}^* = \underline{Q}_{t+\Delta t} - \underline{F}_t^k$$

11. Solve for increments in nodal point temperatures $\Delta\underline{T}_{t+\Delta t}$ using the latest $\underline{K}^*$ matrix:

$$\underline{K}^* \Delta\underline{T}_{t+\Delta t} = \underline{Q}_{t+\Delta t}$$

12. Check if iteration required for heat flow equilibrium:

    NO : Advance to Step 21

    YES : Continue

13. Initialize:

    a. Iteration counter: j = 0

    b. First approximation: $\underline{T}_{t+\Delta t}^{(1)}$

14. Increment iteration counter: j = j + 1

15. Calculate jth approximation to nodal point temperatures and time derivatives of nodal point temperatures:

$$\underline{T}_{t+\Delta t}^{(j)} = \underline{T}_t + \Delta\underline{T}_{t+\Delta t}^{(j)}$$

$$\dot{\underline{T}}_{t+\Delta t}^{(j)} = \frac{1}{\Delta t} (\underline{T}_{t+\Delta t}^{(j)} - \underline{T}_t)$$

TABLE 5.2 (CON'T)

16. Compute the internal heat flow due to conduction at time t + Δt:

$$\underline{F}^{k(j)}_{t+\Delta t} = \underline{K}_t \, \underline{T}^{(j)}_{t+\Delta t}$$

17. Calculate jth out-of-balance heat flow rates:

$$\underline{Q}^{r(j)}_{t+\Delta t} = \underline{Q}_{t+\Delta t} - \underline{C}_t \, \dot{\underline{T}}^{(j)}_{t+\Delta t} - \underline{F}^{k(j)}_{t+\Delta t}$$

18. Solve for jth correction to temperature increments using the latest $\underline{K}^*$ matrix:

$$\underline{K}^* \, \Delta\Delta\underline{T}^{(j)}_{t+\Delta t} = \underline{Q}^{r(j)}_{t+\Delta t}$$

19. Calculate new temperature increments:

$$\Delta\underline{T}^{(j+1)}_{t+\Delta t} = \Delta\underline{T}^{(j)}_{t+\Delta t} + \Delta\Delta\underline{T}^{(j)}_{t+\Delta t}$$

20. Check for convergence:

$$\frac{\| \Delta\Delta\underline{T}^{(j)}_{t+\Delta t} \|_2}{\| \Delta\underline{T}^{(j+1)}_{t+\Delta t} - \underline{T}_t \|_2} < \text{TOL}$$

where the Euclidean norm $\| z \|_2 = \sqrt{\sum_{i=1}^{n} |z_i|^2}$.

If convergence : $\Delta\underline{T}_{t+\Delta t} = \Delta\underline{T}^{(j+1)}_{t+\Delta t}$: Advance to Step 21.

If no convergence and j < NITER : Advance to Step 14. Otherwise, restart using a smaller time step.

21. Calculate new nodal point temperatures:

$$\underline{T}_{t+\Delta t} = \underline{T}_t + \Delta\underline{T}_{t+\Delta t}$$

22. Repeat for the next time step : Advance to Step 1.

6. COMPUTER PROGRAMS

Two finite element analysis programs have been developed for this study. The program DOT ("Determination Of Temperatures") is a general purpose computer program for both linear and nonlinear analysis of transient heat conduction problems and the program DETECT ("DEtermination of TEmperatures in ConsTruction") is a linear heat transfer program for the analysis of structures constructed incrementally (e.g. dams, bridge piers and large foundations). Both of these programs have been developed based on the theory presented in the previous sections of this dissertation and are coded in the standard Fortran IV language. A description of the required input data for the DOT and DETECT programs can be found in Appendices A and B, respectively, along with the Fortran listings.

In order to minimize the storage requirements and optimize efficiency, a compacted storage scheme is used in the DOT and DETECT programs to store the effective system conductivity matrix $\underline{K}^*$. This matrix is stored as a one-dimensional array in which only those elements below the "skyline" of $\underline{K}^*$ are processed. This technique is similar to the one used in the nonlinear structural analysis program NONSAP [30].

The equilibrium equations are solved using the linear equation solver COLSOL [31]. This subroutine uses Gauss elimination on the positive-definite symmetrical system of equations, and requires a minimum number of operations since only those elements within the skyline of the $\underline{K}^*$ matrix are processed.

In order to obtain maximum program capacity, the finite element groups are processed in blocks according to their type (e.g. planar or axisymmetric, convection, radiation or cooling pipes). Secondary or disc storage is then used to store each block of these finite elements. Using this element group data, the effective system thermal conductivity matrix $\underline{K}^*$ is assembled and stored in primary storage (blank common).

The lowest primary storage locations are reserved throughout the solution for the storage of each block of element group data which is read in from secondary storage. The user must supply the maximum estimated number of storage locations (NUMEST) required to store any individual element group in primary storage. During the input phase, the program calculates the exact number of primary storage locations (MAXEST) required for each element group and NUMEST is reset to MAXEST.

To further improve the primary storage requirements, variable dimensioning is used to dynamically allocate the required core storage remaining into a single array A in blank common.

The program DOT has the option of punching the nodal point temperatures onto data cards after every n time steps as specified by the user. This punched temperature history can then be used as input to a thermal stress analysis program which uses the same finite element model as the heat transfer program.

7. SAMPLE ANALYSES

The results of a number of heat transfer problems are presented in this chapter. These analyses were selected to test all major features of the programs DOT and DETECT such as time-dependent boundary conditions, temperature or heat flux constraints, nonzero initial conditions, internal heat generation, surface convection and radiation, nonlinear material properties and incremental construction. Problems involving both steady state and transient response are included, and whenever possible the results have been compared with either analytical results or numerical solutions obtained by other investigators.

7.1 STEADY STATE HEAT CONDUCTION IN A SQUARE PLATE

A 4 by 4-in. square plate, as shown in Fig. 7.1, has one edge maintained at 100^{o}F and the other edges held at 0^{o}F. The steady state temperature distribution obtained used the program DOT and a constant value for thermal conductivity.

The temperature distribution was first obtained using a coarse 4 x 4 finite element grid connected by 25 nodal points. The analysis was then repeated using a finer 8 x 8 grid containing 81 nodes. In the second solution, it was only necessary to use a 4 x 8 grid with 45 nodes to represent the half-width of the plate since the global plane (Y = 2 in.) is a plane of conduction symmetry.

The exact solution for the temperature at any point (x,y) within the plate is given by Carslaw and Jaeger [4] as the series solution:

$$T(x,y) = \frac{4T_o}{\pi}\sum_{n=0}^{\infty}\frac{1}{(2n+1)}\sin(2n+1)\frac{\pi x}{a}\sinh(2n+1)(a-y)\frac{\pi}{a}\,\mathrm{cosech}(2n+1)\pi$$

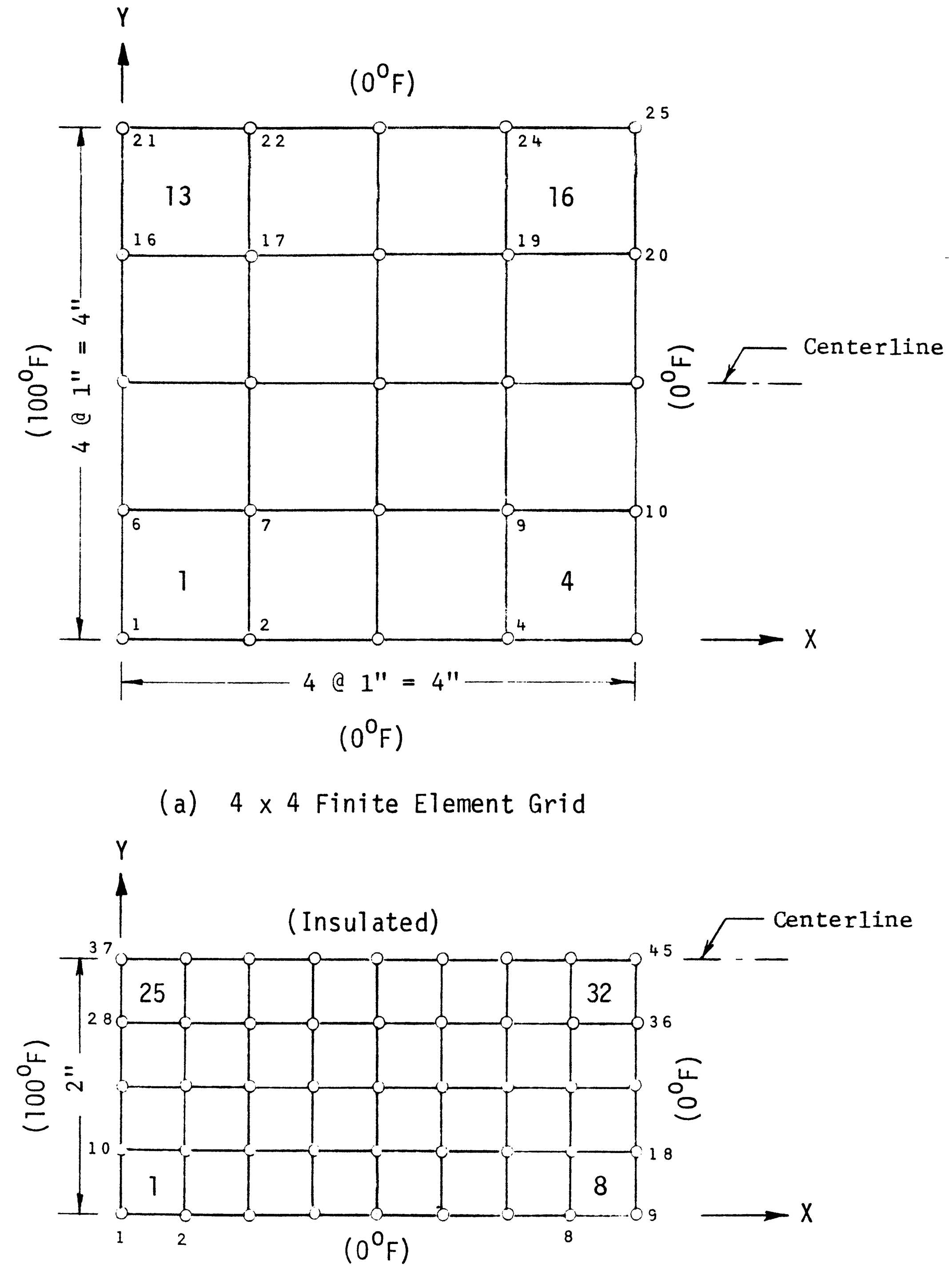
where

a = length of one side of the plate (4 inches)

T_o = temperature of side $Y = 0$ (100^oF)

x,y = coordinate system defined in Fig.

The analytical solution for the temperature distribution along the centerline ($Y = 2$ in.) of the plate is compared in Fig. 7.2 with the DOT solution. As can be seen, the agreement is excellent and the solution improved when the finer 8 x 8 mesh was used. The results of the numerical solution also matched those obtained numerically by Peterson [25].



(a) 4 x 4 Finite Element Grid

(b) 4 x 8 Finite Element Grid

FIG. 7.1 FINITE ELEMENT MODELS OF 4 BY 4-IN. SQUARE PLATE

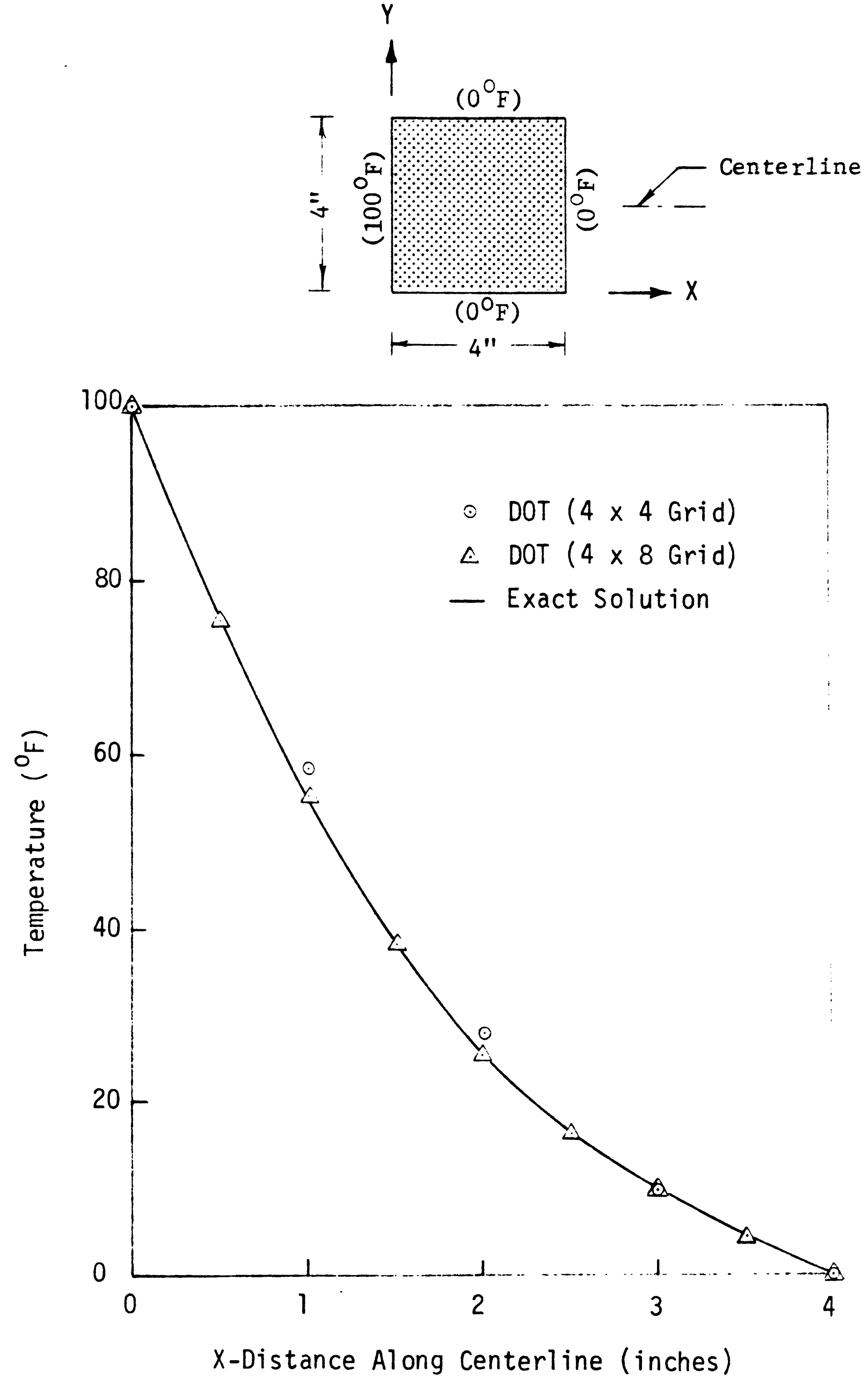


FIG. 7.2 TEMPERATURE DISTRIBUTION ALONG CENTERLINE OF THE 4 BY 4-INCH SQUARE PLATE

7.2 STEADY STATE HEAT CONDUCTION IN A LONG HOLLOW CYLINDER

A long hollow cylinder is subjected to an inside temperature of 100^{o}F and an outer surface temperature of 0^{o}F. The cylinder is insulated to prevent axial heat flow and has an inside radius of one inch and an outer radius of two inches. The program DOT was used to obtain the steady state temperature distribution in the radial direction, assuming a constant value of thermal conductivity throughout the cylinder.

For the finite element model, a 1-in. length of the cylinder is represented using ten equally spaced 4-node elements with 22 nodal points through the 1-in. thickness of the cylinder wall. This model is shown in Fig. 7.3(a).

The temperature at any arbitrary radius r within the curved cylinder wall is given by Kreith [29] as:

$$T(r) = T_i - \frac{T_i - T_o}{\ln(r_o/r_i)} \ln(\frac{r}{r_i})$$

where

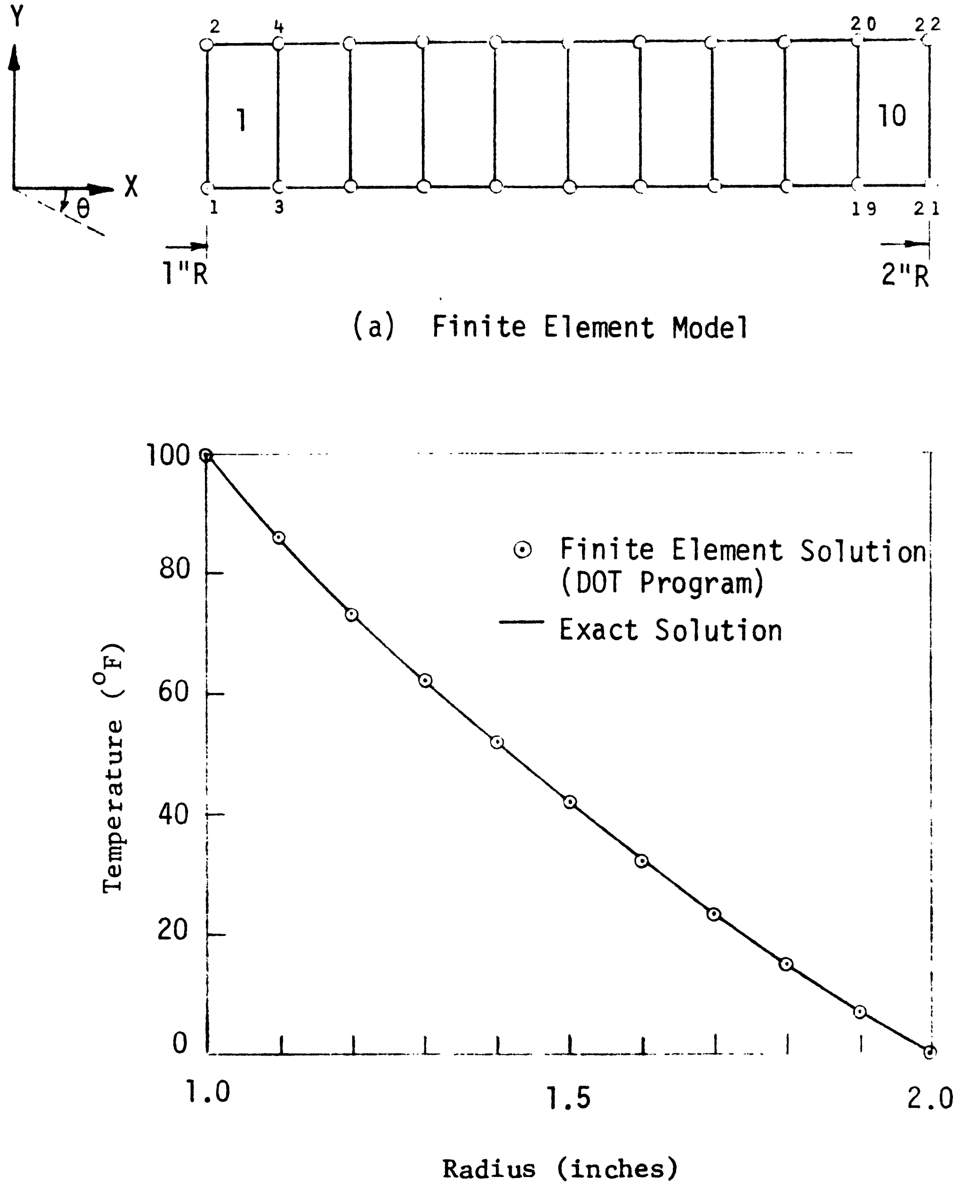
T_i = inner surface temperature

T_o = outer surface temperature

r_i = inside radius

r_o = outside radius

The steady state temperature profile through the cylinder wall obtained using DOT was found to closely match the exact solution given above, as shown in Fig. 7.3(b).



(a) Finite Element Model

(b) Temperature Distribution In Radial Direction

FIG. 7.3 STEADY STATE HEAT CONDUCTION IN A LONG HOLLOW CYLINDER

7.3 STEADY STATE HEAT CONDUCTION IN A RECTANGULAR CONCRETE COLUMN EXPOSED TO CONVECTION BOUNDARY CONDITIONS

An exposed 14 by 36-inch rectangular concrete column is subjected to an outside weather temperature of 100°F and an inside environment temperature of 0°F. The inside and outside surfaces of the column are separated by an 8-inch thick wall.

The convection coefficient for the 100°F environment is 0.5 BTU/hr-ft^2-°F and for the 0°F environment is 6.0 BTU/hr-ft^2-°F. Both coefficients decrease linearly to zero from the surface to the center of the abutting wall, as indicated in Fig. 7.4. The thermal conductivity of the concrete is assumed to have a uniform value of K = 1.0 BTU/hr-ft-°F.

In constructing a finite element model, it is only necessary to consider one-half of the column cross-section ($Y \leq 7$ in.) since the column centerline at Y = 7 in. is a plane of conduction symmetry.

Ninety-six 4-node planar elements containing 125 nodes are used to represent a one inch deep transverse section of the column, as shown in Fig. 7.4. Nodes lying in the Y = 7 in. plane are taken to be insulated, and the element surfaces in the X = 0, Y = 0 and X = 36 in. planes are exposed to convective surface heat transfer.

The program DOT was used to obtain two steady state temperature distributions, one along the column face (Y = 0) and the other along the column centerline (Y = 7 in.). The results are compared in Fig. 7.5 with a Fourier analysis solution performed by Peavy [32]. It can be seen that the temperature distributions obtained using the DOT program closely stradle Peavy's solution, which predicted the mean temperature over the entire 14-in. width of the cross-section.

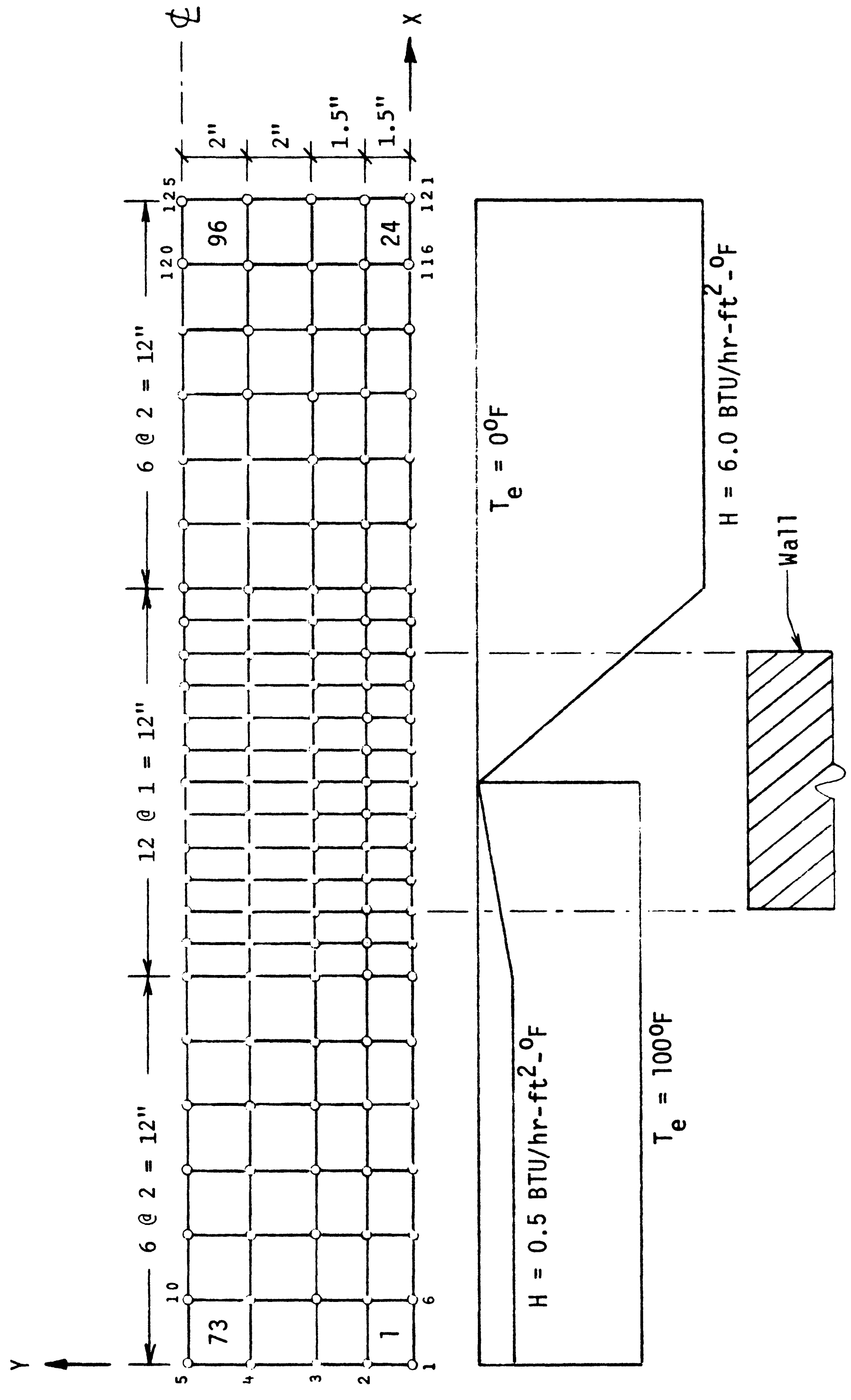


FIG. 7.4 FINITE ELEMENT MODEL OF THE 14 BY 36-IN. RECTANGULAR CONCRETE COLUMN

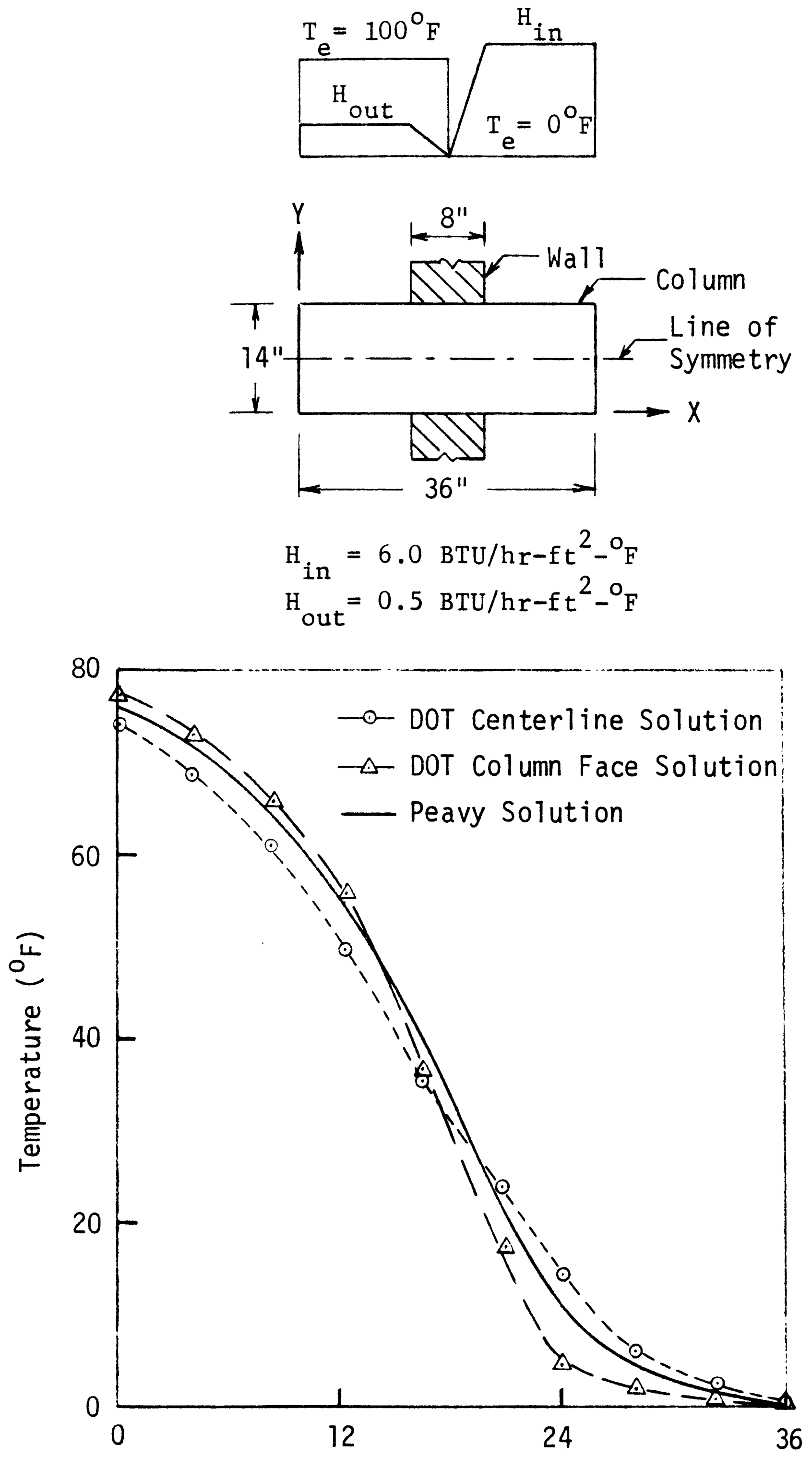


FIG. 7.5 TEMPERATURE DISTRIBUTIONS OVER THE 36-IN. WIDTH OF THE COLUMN

7.4 INTERNAL HEAT GENERATION IN A PARALLEL-SIDED SLAB

An 8-inch thick slab, infinite in extent, is initially at 0^{o}F. At time $t = 0^{+}$, heat is generated internally in the slab at a uniformly distributed rate $\dot{q}$ = 2000 BTU/sec-in^{3}. The external surfaces of the slab are maintained at 0^{o}F for all time.

The material parameters chosen for this study are:

Thermal Diffusivity κ = 16 in^{2}/sec

Specific Heat c = 1.0 BTU-in/sec^{2}-lb-^{o}F

Mass Density ρ = 1.0 sec^{2}-lb/in^{4}

which are equivalent to a thermal conductivity value of K equal to 16.0 BTU/sec-in-^{o}F ($\kappa = K/\rho c$).

In developing a finite element model for the slab, the slab centerline is recognized as a plane of conduction symmetry. Ten equally spaced 4-node elements are used to represent a one square inch section through the 4-in. half-depth of the slab, as shown in Fig. 7.6(a). Nodes 21 and 22 at the surface (X = 4 in.) have a prescribed temperature boundary condition of 0^{o}F and the nodes 1 and 2 at the slab mid-depth (X = 0) are assumed to be insulated for all values of time.

The exact solution for the temperature at time t at any depth x within a slab subjected to internal heat generation is given by Carslaw and Jaeger [4] as:

$$T(x,t) = \frac{\dot{q}L^{2}}{2K}\left\{1-\left(\frac{x}{L}\right)^{2} - \frac{32}{\pi^{3}}\sum_{n=0}^{\infty}\frac{(-1)^{n}}{(2n+1)^{3}}\cos\left[\frac{(2n+1)\pi x}{2L}\right] e^{-\kappa\left[\frac{(2n+1)\pi}{2L}\right]^{2} t}\right\}$$

where

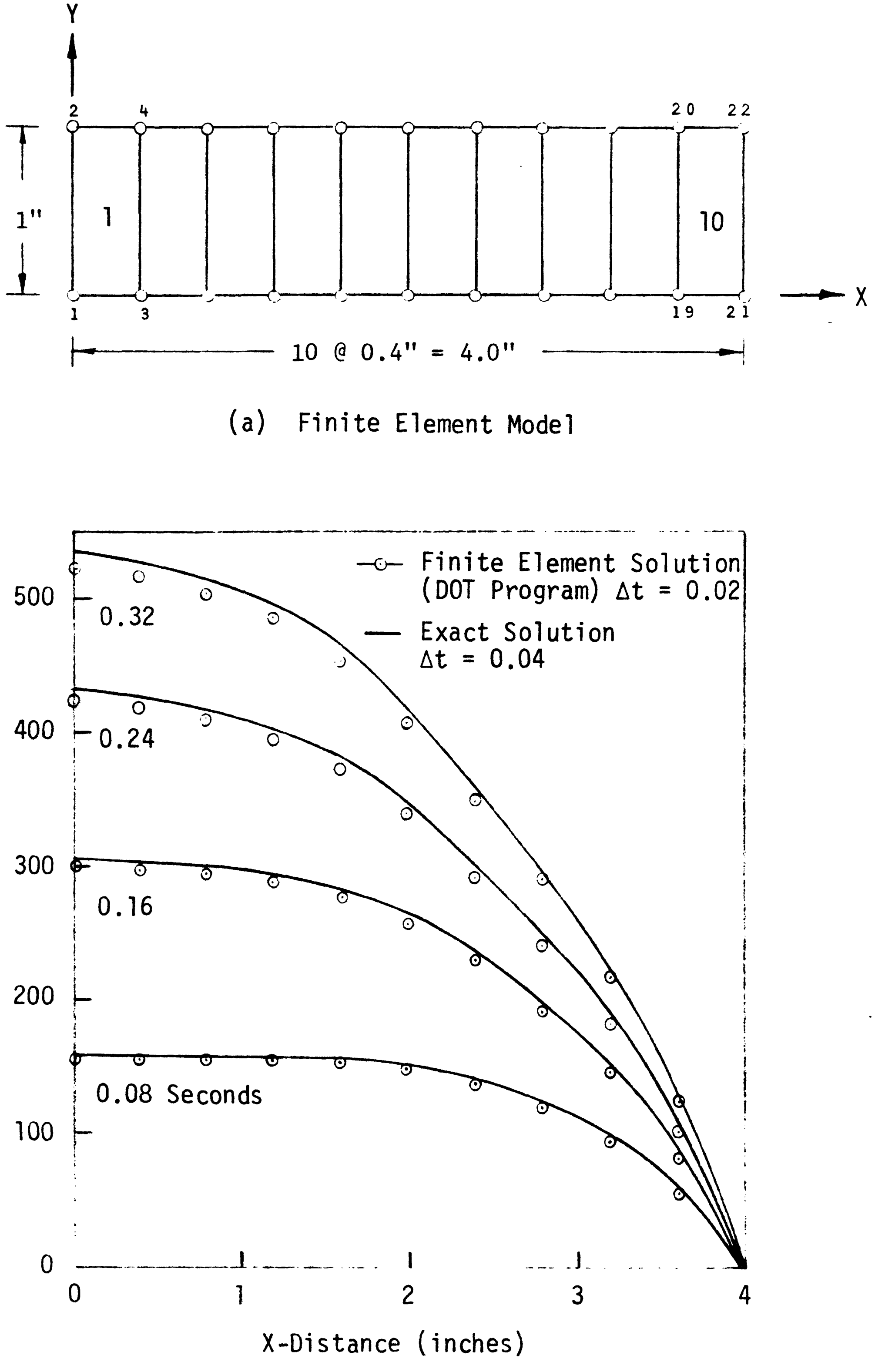
$\dot{q}$ = rate of internal heat generation (2000 BTU/sec-in^3)

L = half-depth of slab (4.0 in.)

K = thermal conductivity (16.0 BTU/sec-in-^{o}F)

κ = thermal diffusivity (16.0 in^2/sec)

Figure 7.6(b) shows the temperature profiles across the half-thickness of the slab. The four curves represent the series solution for the temperature distribution at times of 0.08, 0.16, 0.24 and 0.32 seconds and the plotted points are the nodal temperatures predicted by the DOT program for these respective times, using a solution time step Δt = 0.02 seconds. It is noted that the finite element solution accurately predicts the temperatures resulting from internal heat generation in the slab.



(a) Finite Element Model

(b) Temperature Distribution

FIG. 7.6 INTERNAL HEAT GENERATION IN A PARALLEL-SIDED SLAB

7.5 TRANSIENT HEAT CONDUCTION IN A SEMI-INFINITE SOLID SUBJECTED TO A UNIT SURFACE HEAT FLUX

A semi-infinite solid initially at zero temperature is exposed to a constant surface heat flux of unit intensity at time $t = 0^+$, i.e. $Q(0,t) = 1.0$, $t > 0$. The material properties are all assumed to be linear (temperature independent) and are assigned unit values:

Conductivity $K = 1.0$ BTU/sec-in-$^{\circ}$F

Specific Heat $c = 1.0$ BTU-in/sec^2-lb-$^{\circ}$F

Mass Density $\rho = 1.0$ sec^2-lb/in^4

which are equivalent to a value of thermal diffusivity κ equal to 1.0 in^2/sec ($\kappa = K/\rho c$).

Fifteen equally spaced 4-node planar elements are used in the analysis to represent a 3 inch depth of the solid, as shown in Fig. 7.7(a). The surface of the solid coincides with the global Y-Z plane. Nodes 31 and 32 on the interior plane (X = 3 in.) are assumed to be insulated and nodes 1 and 2 on the surface of the solid (X = 0) are subjected to the unit value of heat flux. Temperatures were computed using the DOT program with two different time steps, namely $\Delta t_1 = 0.025$ and $\Delta t_2 = 0.05$ seconds.

The exact solution for the temperature at time t and at any depth x within the solid subjected to a unit surface heat flux is given by Carslaw and Jaeger [4] as:

$$T(x,t) = \frac{2}{K}\left\{\left(\frac{\kappa t}{\pi}\right)^{1/2} e^{-x^2/4\kappa t} - \frac{x}{2}\,\mathrm{erfc}(x/2\sqrt{\kappa t})\right\}$$

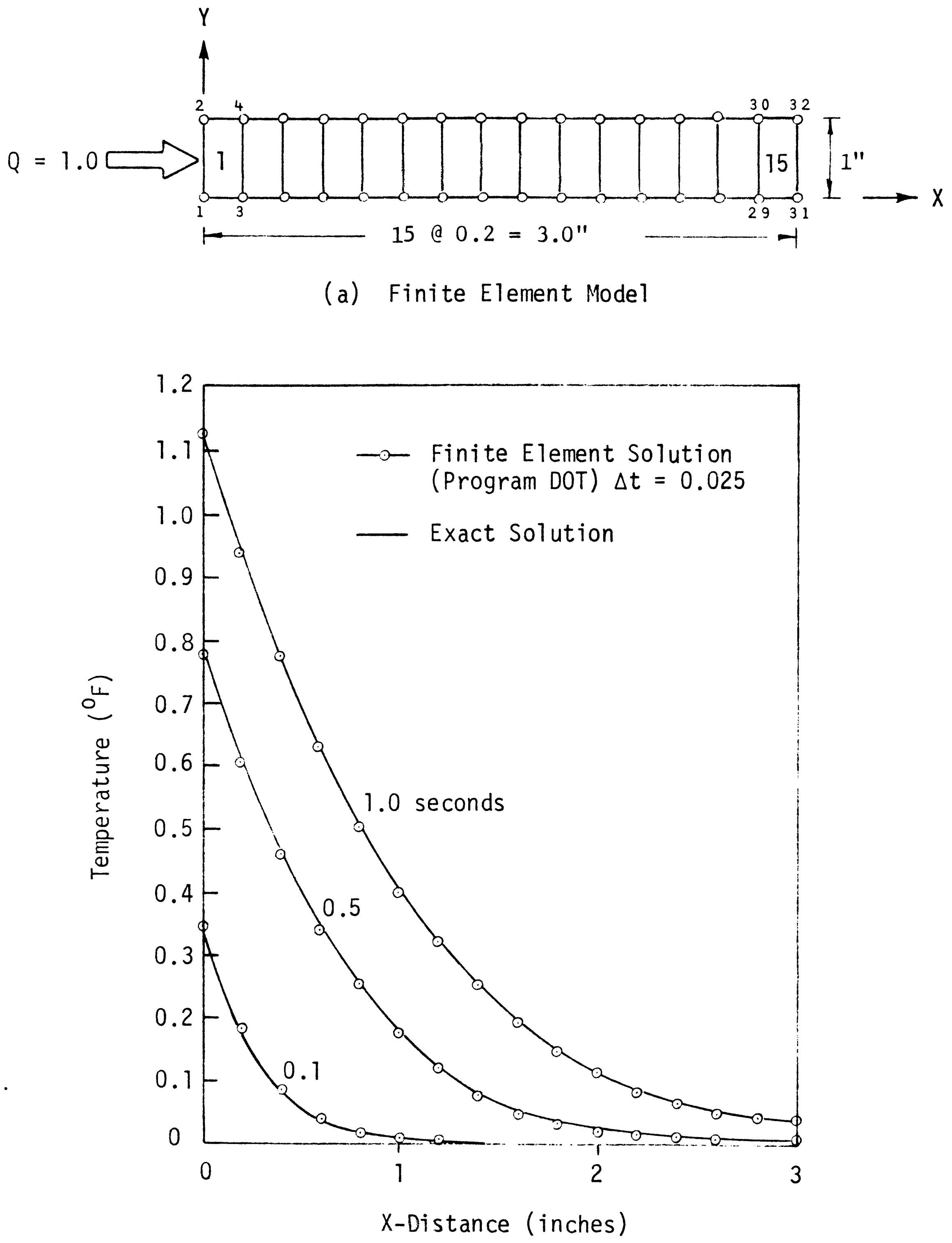
where

K = thermal conductivity (1.0 BTU/sec-in-^{o}F)

κ = thermal diffusivity (1.0 in^2/sec)

The temperature distributions into the depth of the solid are shown in Fig. 7.7(b) for time values of 0.1, 0.5 and 1.0 seconds. The solid curves represent the exact solution and the symbol points are the nodal temperatures obtained with the DOT program and the first solution time step Δt_1 = 0.025 seconds. As can be seen, the agreement is very close.

Fig. 7.8 is a plot of the surface temperature of the solid versus time from the start of heating. The symbol points represent the finite element solution for the two different time steps. Both solutions are seen to accurately predict the temperature response of the solid.



(a) Finite Element Model

(b) Temperature Distribution

FIG. 7.7 SEMI-INFINITE SOLID SUBJECTED
TO A UNIT SURFACE HEAT FLUX

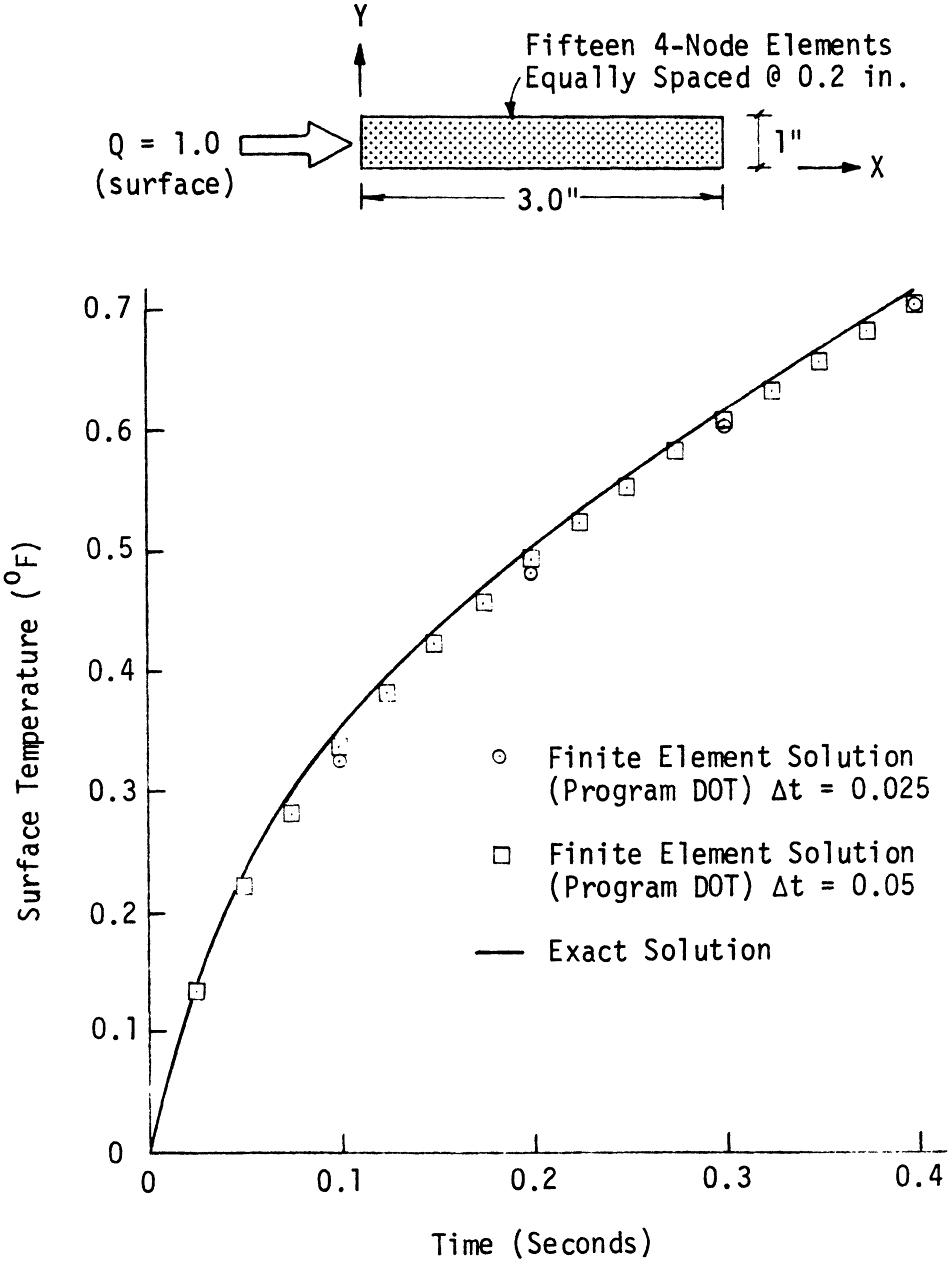


FIG. 7.8 SURFACE TEMPERATURE OF A SEMI-INFINITE SOLID VS. TIME FROM START OF HEATING

7.6 TRANSIENT HEAT CONDUCTION IN A FINITE LENGTH ROD

A 0.5-inch long rod with a cross-sectional area of one square inch has an initial temperature distribution which varies linearly with distance (x) along its length:

$$T(x,0) = T_0(x/L)$$

where

$$T_0 = 100^0F$$

$$L = 0.5 \text{ inches}$$

At time $t = 0^+$, the end $x = 0.0$ is maintained at 0^0F and the end $x = 0.5$ in. is insulated $(Q = 0)$.

The material parameters chosen for this study are:

Conductivity $K = 6.0$ BTU/sec-in-^0F

Specific Heat $c = 3.0$ BTU-in/sec^2-lb-^0F

Mass Density $\rho = 2.0$ sec^2-lb/in^4

which are equivalent to a thermal diffusivity value κ equal to 1.0 in^2/sec $(\kappa = K/\rho c)$.

For the finite element model shown in Fig. 7.9(a), ten 4-node planar elements are used along the length of the rod. Nodes 1 and 2 are maintained at 0^0F for all values of time and nodes 21 and 22 are insulated (zero external heat flow). Temperatures were obtained using the DOT program with a time step $\Delta t = 0.001$ seconds.

The exact solution for the temperature at any distance x along the rod at time t is given by the series solution [33]:

$$T(x,t) = \frac{8T_0}{\pi^2} \sum \frac{(-1)^n}{(2n+1)^2} \sin\left[\frac{(2n+1)\pi x}{2L}\right] e^{-\kappa \left[\frac{(2n+1)\pi}{2L}\right]^2 t}$$

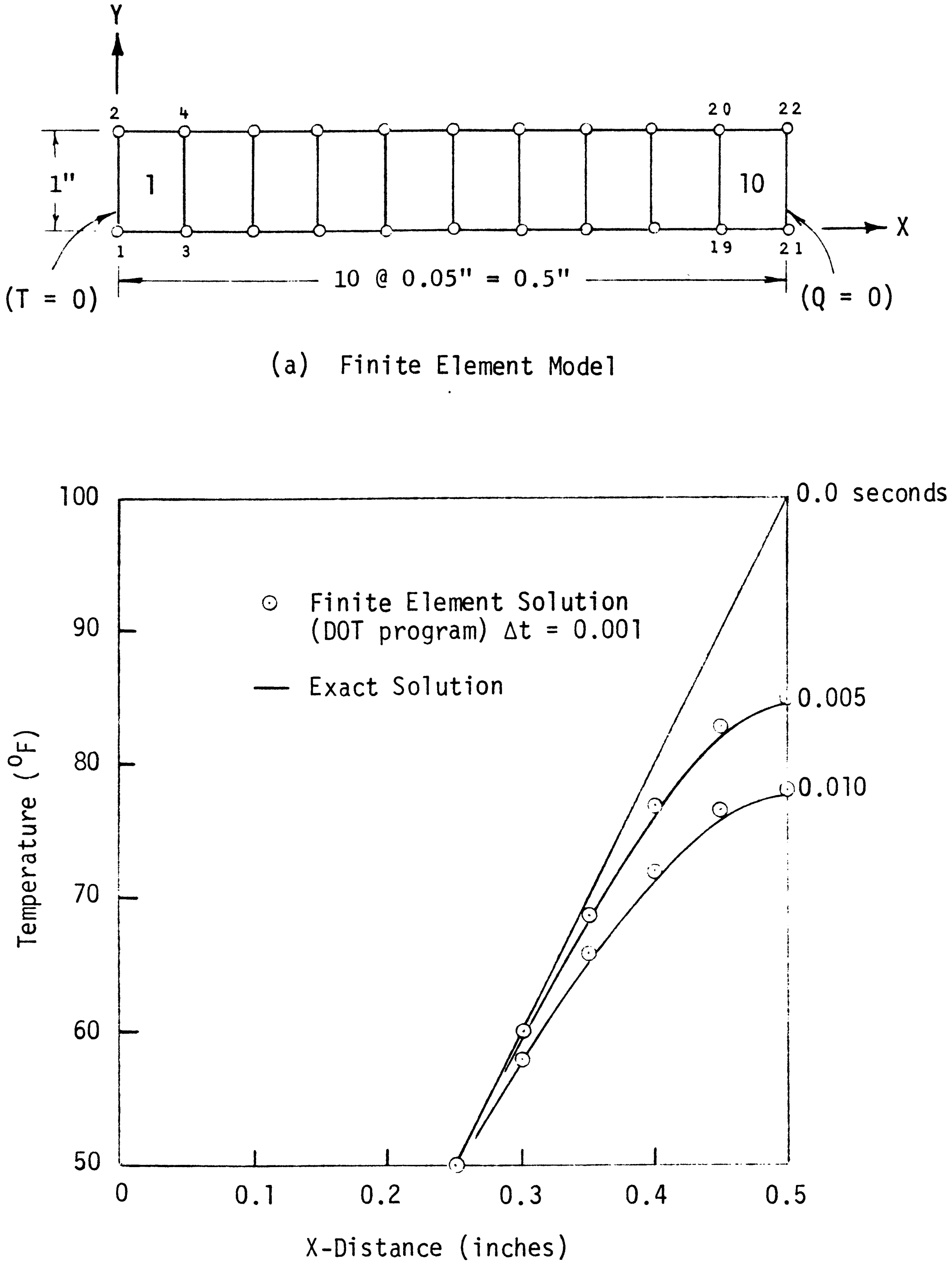
where

T_o = temperature at end x = 0.0 (100^oF)

L = length of rod (0.5 in.)

κ = thermal diffusivity (1.0 in^2/sec)

A comparison between the finite element and exact solution results is shown in Fig. 7.9(b). The solid curves represent the series solution for the temperature distribution along the rod at times of 0.005 and 0.010 seconds and the plotted points are the nodal temperatures predicted by the DOT program for these respective times. It can be seen that the numerical finite element results are in close agreement with the exact solution.



(a) Finite Element Model

(b) Temperature Distributions

FIG. 7.9 TRANSIENT HEAT CONDUCTION IN A FINITE LENGTH ROD

7.7 TRANSIENT HEAT CONDUCTION IN AN INFINITE PARALLEL-SIDED SLAB

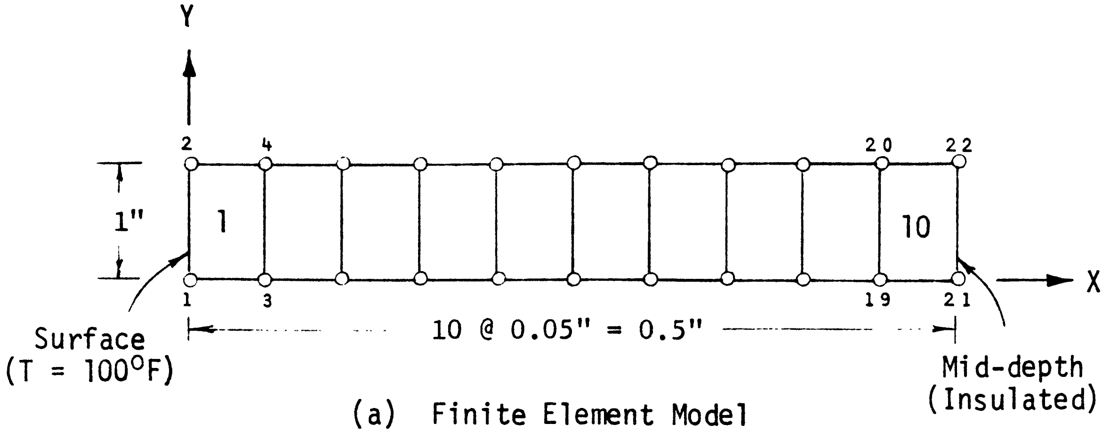
A one-inch thick slab, infinite in extent, is initially at 0^{o}F. At time $t = 0^{+}$, both faces of the slab are raised to 100^{o}F and held constant for all time.

The material parameters chosen for the slab are:

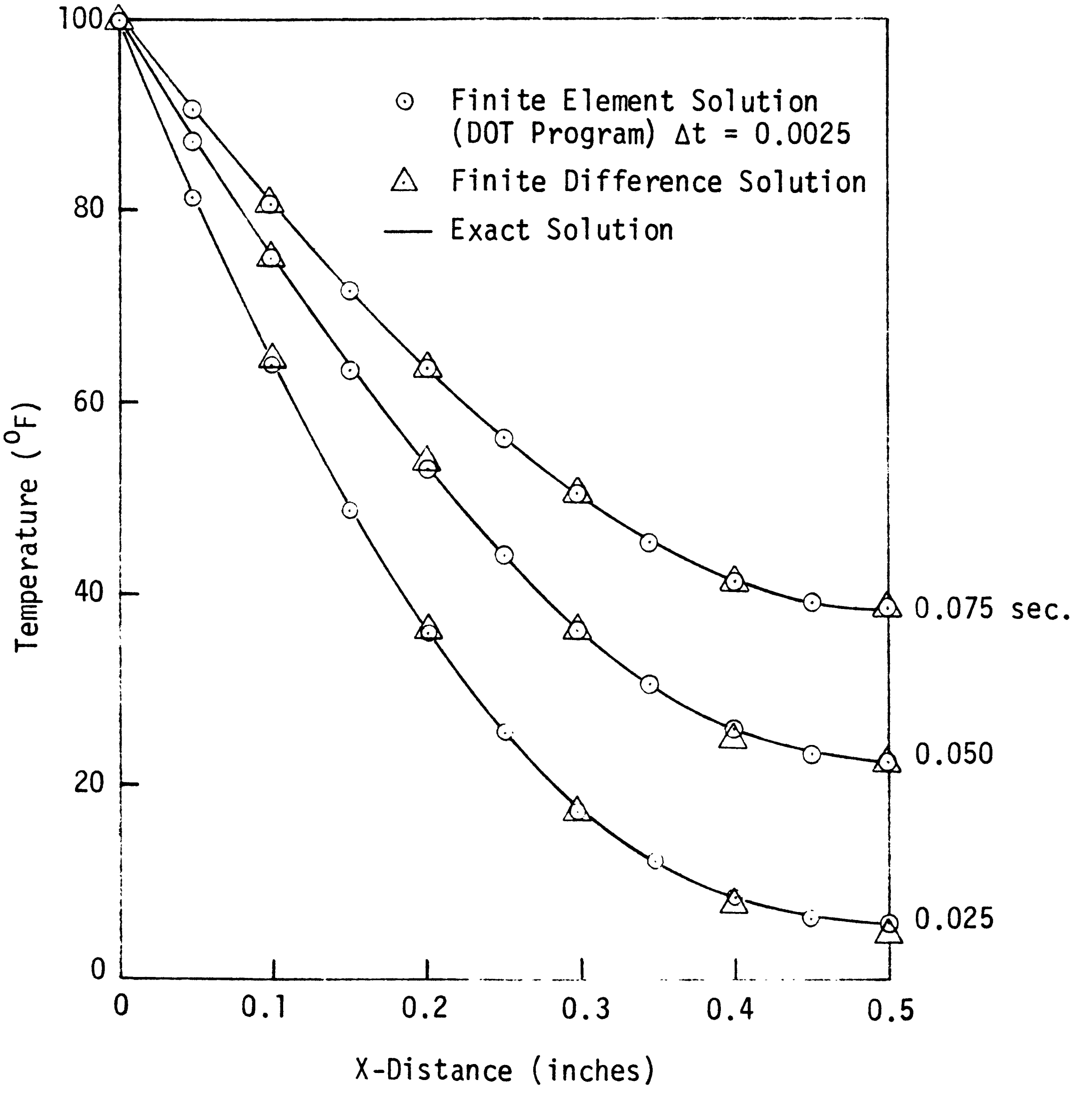
Conductivity $K = 6.0$ BTU/sec-in-^{o}F
Specific Heat $c = 2.0$ BTU-in/sec^{2}-lb-^{o}F
Mass Density $\rho = 3.0$ sec^{2}-lb/in^{4}

For the finite element model, ten equally spaced 4-node planar elements are used to represent a one square inch section through the 0.5-in. half-depth of the slab as shown in Fig. 7.10(a). Nodes 1 and 2 on the surface of the slab are maintained at 100^{o}F for time $t > 0$ and nodes 21 and 22 at the slab mid-depth lie in a plane of conduction symmetry and are treated as insulated nodes.

Figure 7.10(b) compares the temperature distributions across the half-thickness of the slab at times of 0.025, 0.050 and 0.075 seconds. The finite element solution was obtained using the program DOT with a time step $\Delta t = 0.0025$ seconds. The finite difference solution points are from Carnahan, Luther and Wilkes [34], and the exact solution is an infinite series expansion given by Olson and Schultz [35]. All three solutions are seen to provide very close agreement in predicting the temperatures through the slab.



(a) Finite Element Model



(b) Temperature Distributions

FIG. 7.10 TRANSIENT HEAT CONDUCTION IN A PARALLEL-SIDED SLAB

7.8 NONLINEAR TRANSIENT HEAT CONDUCTION THROUGH AN INFINITE PARALLEL-SIDED SLAB

A 20-inch thick slab, infinite in extent, is initially at a reference temperature of 100^{o}F as shown in Fig. 7.11. At time $t = 0^{+}$, the temperature on the left face of the slab ($X = 0$) is raised to 200^{o}F and maintained at this level until time $t = 10$ seconds when it drops back to 100^{o}F. The thermal conductivity K is assumed to vary linearly with temperature and the specific heat c is taken to be constant. The material parameters chosen for this study are:

Conductivity $K = 2 + (0.01)T$ BTU/sec-in-^{o}F

Specific Heat $c = 8$ BTU-in/sec^{2}-lb-^{o}F

Mass Density $\rho = 0.25$ sec^{2}-lb/in^{4}

For the finite element model, twenty equally spaced 4-node planar elements are used to represent a one square inch section through the 20-in. thickness of the slab. The transient temperature distributions were obtained using the program DOT with a time step $\Delta t = 1$ second and reforming the system equations every time step.

This same problem was also solved numerically by Aguirre-Ramirez and Oden [16] and by Wilson, Bathe and Peterson [24]. Wilson et al. used an algorithm involving heat flow equilibrium iteration and performed three analyses in which new conductivity matrices were formed and triangularized (1) in each time step, (2) every 5th time step and (3) only once at the start of solution when $T = 100^{o}$F. The temperatures calculated in these three analyses were found to be the same within 5 digits of accuracy, and the average number of iterations required in the analyses were 2, 3 and 5 respectively.

Table 7.1 compares the results obtained from the DOT analysis with those obtained by Wilson et al. for the temperature at a distance x from the left face of the slab at times t = 10 and 11 seconds. The two solutions are seen to be in close agreement. Additionally, both sets of results in Table 7.1 compare favorably with the predicted temperature variation given in a small figure in Ref. [16].

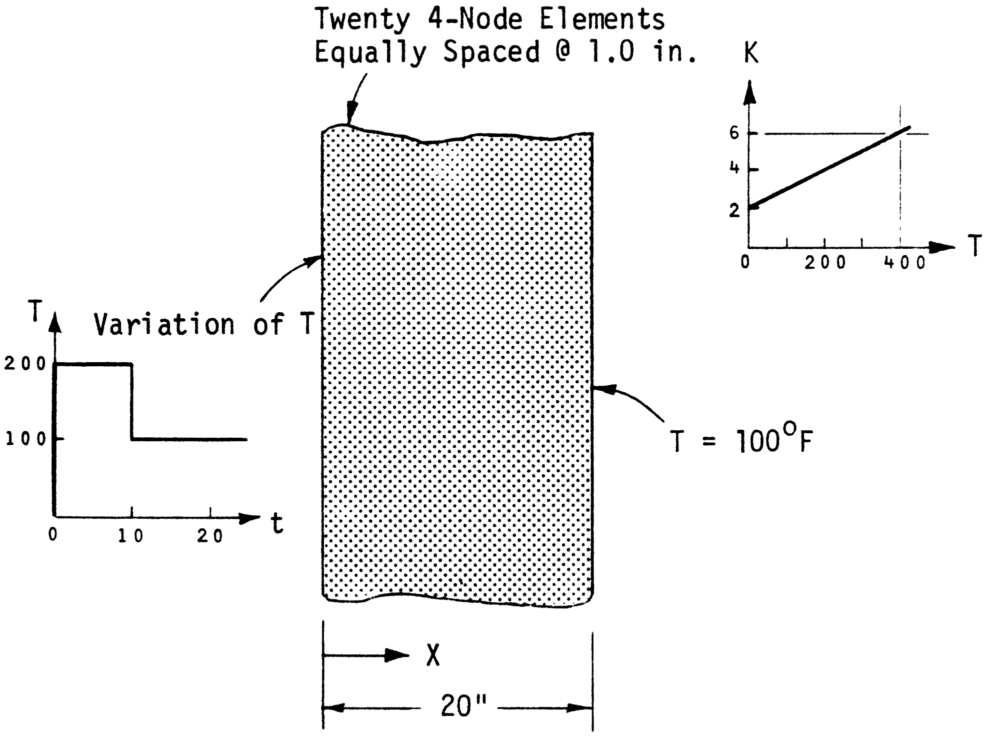


FIG. 7.11 NONLINEAR TRANSIENT HEAT CONDUCTION THROUGH AN INFINITE PARALLEL-SIDED SLAB

TABLE 7.1: TEMPERATURES AT A DISTANCE X FROM THE LEFT FACE OF THE SLAB AT TIMES t = 10 AND t = 11 SECONDS

Distance x	T @ t = 10		T @ t = 11	
	DOT Solution	Wilson Solution	DOT Solution	Wilson Solution
0	200.00	200.00	100.00	100.00
1	187.23	187.35	137.80	141.52
2	174.41	174.68	150.89	153.46
3	161.95	162.38	151.65	153.19
4	150.27	150.83	146.67	147.68
5	139.71	140.38	139.42	140.23
6	130.55	131.26	131.80	132.52
7	122.90	123.59	124.74	125.41

Note: All temperature values are in ^{o}F

7.9 TRANSIENT TEMPERATURE ANALYSIS OF A MASS CONCRETE STRUCTURE CONSTRUCTED INCREMENTALLY

This example was selected to illustrate the capability of the finite element program DETECT to predict the temperature distribution in a mass concrete structure constructed incrementally. The structure chosen for this study is the Norfork dam, a concrete gravity dam [36]. The basic geometry of the section of dam analyzed is shown in Fig. 7.12. The finite element representation of the dam cross-section is shown in Fig. 7.13. This model includes a 40-ft. portion of the foundation rock, and the element boundaries in the concrete mass coincide with the theoretical lift placement boundaries used in this study.

The concrete temperature in the dam will increase rapidly above the placement temperature due to the heat given off by the hydrating cement. The degree of temperature rise will depend on the type of cement used, the quantity of cement in the mixture, the diffusivity of the aggregate used and the construction scheme. Most dams are typically constructed in 5 to 7-1/2 ft. high lifts placed at 5 to 7 days apart, but for the purposes of this example, the dam is idealized as being constructed in twelve 20-ft. high lifts placed at 30 day intervals, for a total construction time of 360 days. The placement temperature of the concrete was taken to be 45^{o}F.

The hypothetical rate of internal heat generation for the Norfork dam concrete shown in Fig. 7.14 was taken from the adiabatic temperature rise data determined for the Oroville dam mass concrete mix no. 3-B(II) containing 4 sacks of cement per cubic yard [37].

The boundary conditions assumed in the analysis are as follows:

(1) At any level of construction, all existing concrete surfaces are exposed to linear, free convection. The convection coefficient selected was 140 BTU/hr-ft^2-^{o}F and the hypothetical 360-day air temperature cycle used in the analysis is shown in Fig. 7.15.

(2) The base of the foundation rock, nodes 1 to 13 in Fig. 7.13, is insulated and maintained as a temperature boundary condition of 55^{o}F for all time.

The material properties for the concrete and foundation rock were taken to be the same, and the following values were selected for this analysis:

Conductivity K = 1.2 BTU/hr-ft-^{o}F
Specific Heat c = 0.22 BTU/lb-^{o}F
Mass Density ρ = 155 lb/ft^3

Fig. 7.16 is a plot of isothermal contours in the dam at the completion of construction (t = 360 days). A similar set of contours is plotted in Fig. 7.17 showing the temperature distribution in the dam one year after the end of construction (t = 720 days). The temperatures in the central core of the dam are observed to be significant (>100^{o}F) at the end of construction (Fig. 7.16) and can be seen to have decreased in magnitude one year later (Fig. 7.17).

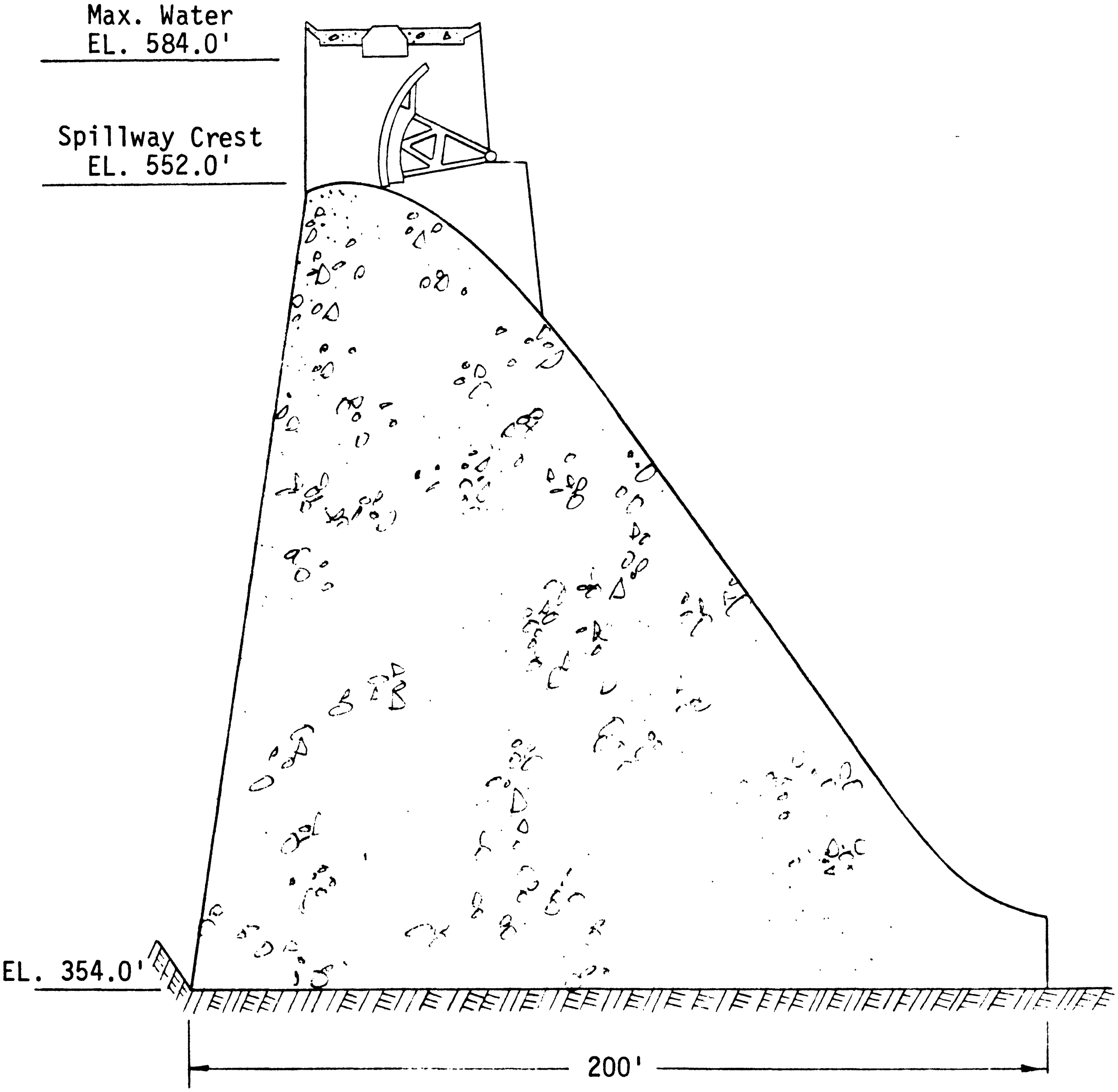


FIG. 7.12 CROSS SECTION OF NORFORK DAM

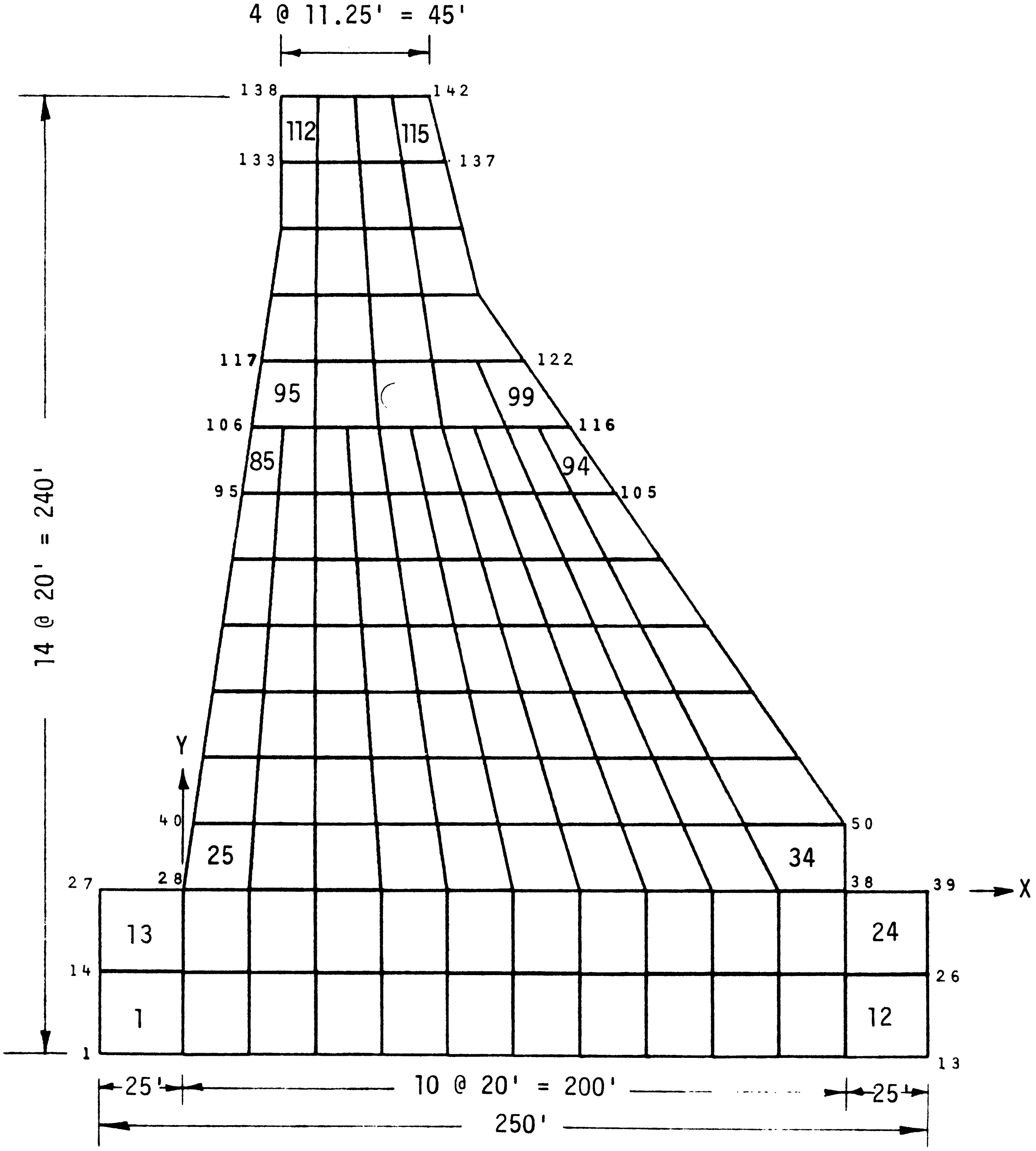
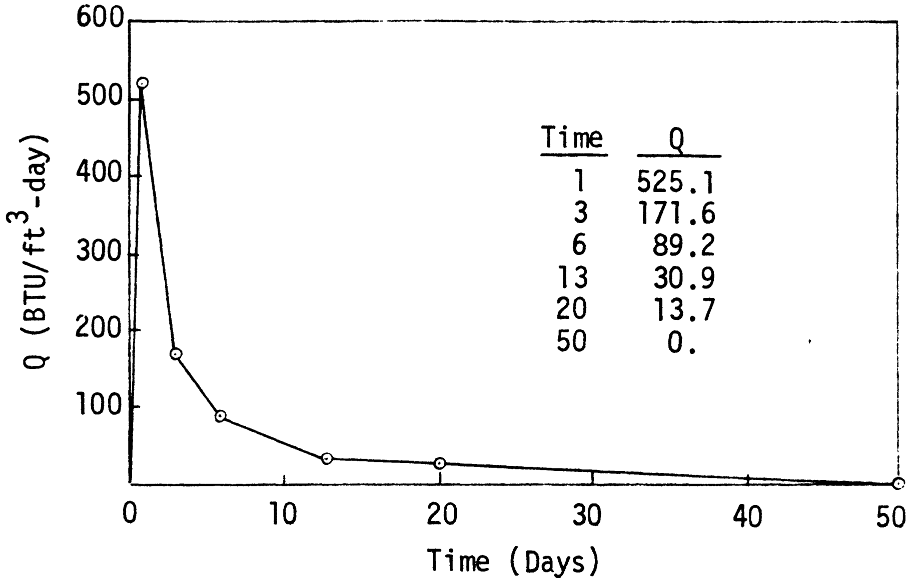


FIG. 7.13 FINITE ELEMENT MESH OF NORFORK DAM



Time	Q
1	525.1
3	171.6
6	89.2
13	30.9
20	13.7
50	0.

FIG. 7.14 HYPOTHETICAL RATE OF INTERNAL HEAT GENERATION FOR THE NORFORK DAM CONCRETE

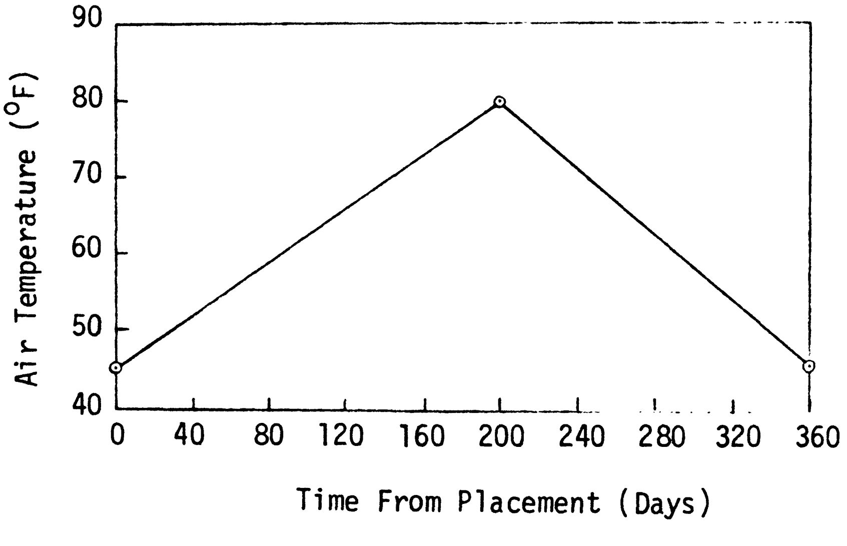


FIG. 7.15 360-DAY HYPOTHETICAL AIR TEMPERATURE CYCLE FOR NORFORK DAM

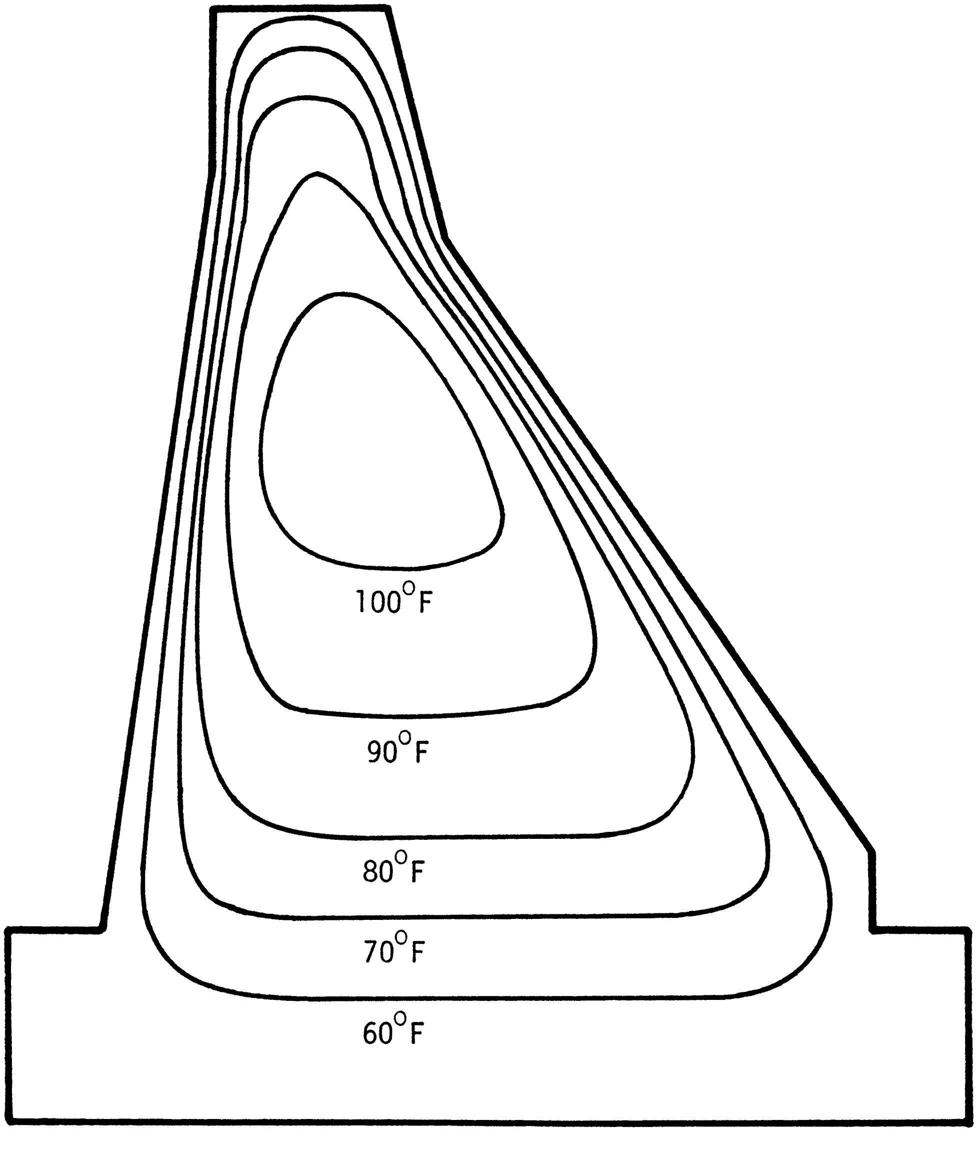


FIG. 7.16 TEMPERATURE CONTOURS IN DAM AT THE END OF CONSTRUCTION (360 DAYS)

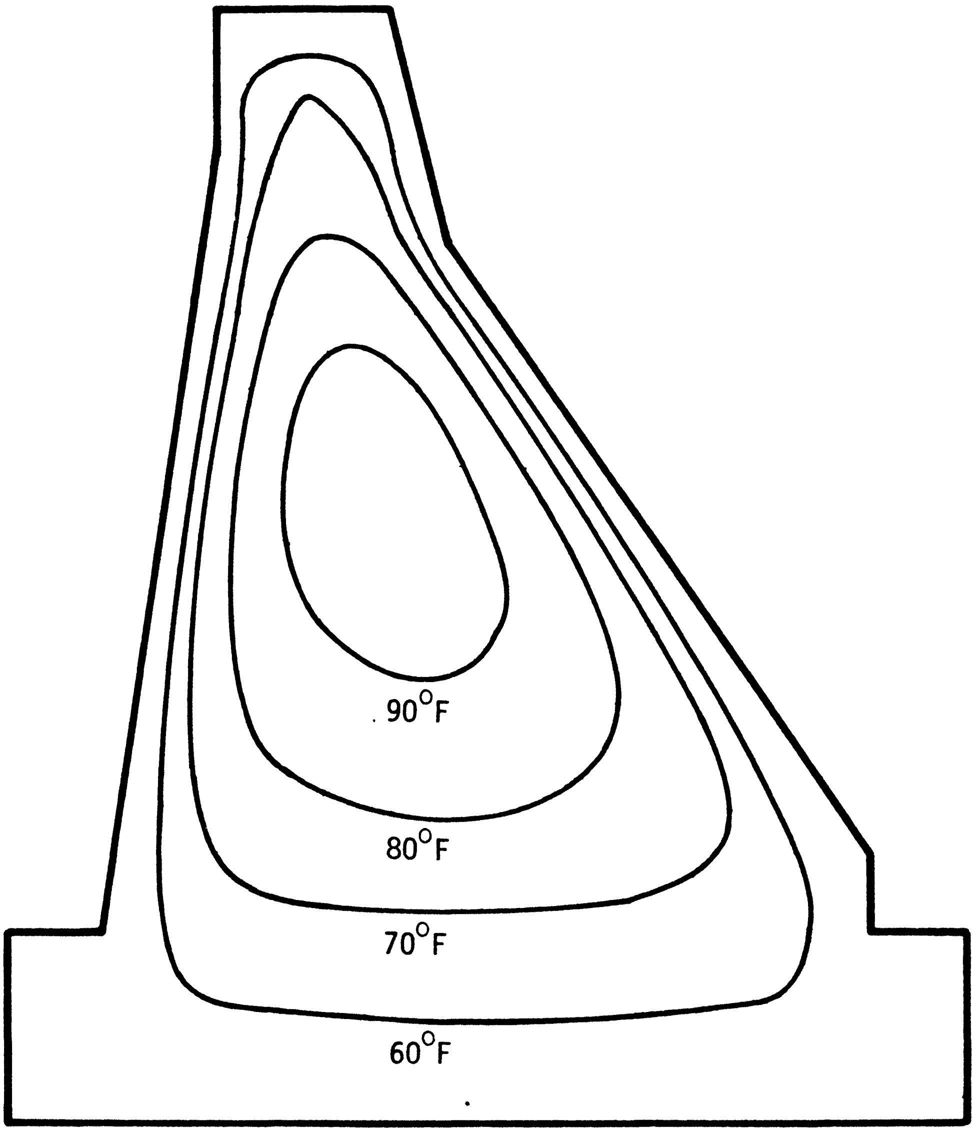


FIG. 7.17 TEMPERATURE CONTOURS IN DAM ONE YEAR FROM THE END OF CONSTRUCTION (720 DAYS)

7.10 NONLINEAR HEAT CONDUCTION IN A SQUARE CONCRETE COLUMN UNIFORMLY EXPOSED TO ASTM FIRE

A 20 x 20-in. plain concrete column, shown in Fig. 7.18(a), is at an initial room temperature of 68°F. At time $t = 0^+$, the column is exposed to a long duration moderately intense fire as defined by ASTM Standard E-119 [38]. The temperature variation of this pseudo-fire with time is given in Table 7.2.

The fire boundary condition on the external column surface is treated as a combination of a linear convection and a nonlinear radiation boundary condition. The convection coefficient (H) chosen for this study is 0.27 BTU/hr-ft^2-°F, the view factor (V) is 1.0, the concrete surface emissivity (ε_s) is 0.9 and the fire emissivity (ε_f) is 0.6.

The thermal conductivity K is assumed to vary with temperature and the specific heat c is taken to be constant. The material properties chosen for this study are those given by Bizri [39]:

(a) Conductivity

$68^\circ F < T < 390^\circ F$	$K = 1.01$ BTU/hr-ft-°F
$390^\circ F < T < 1650^\circ F$	$K = [1.01 - 0.0004(T-390)]$ BTU/hr-ft-°F
$1650^\circ F < T$	$K = 0.506$ BTU/hr-ft-°F

(b) Specific Heat $c = 0.272$ BTU/lb-°F

(c) Density $\rho = 150$ lb/ft^3

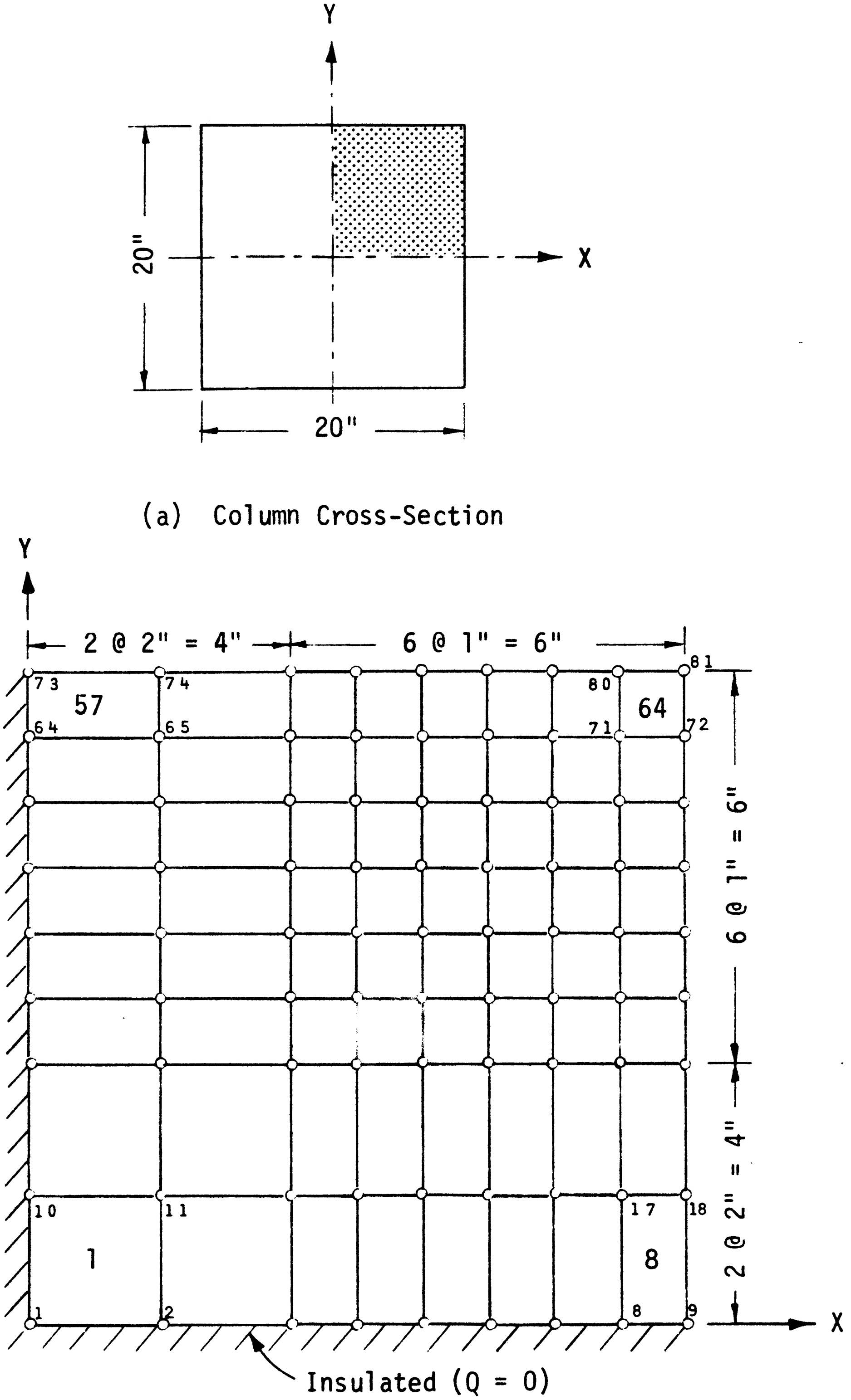
In developing a finite element model, it is only necessary to consider one quadrant of the column in the analysis because the column is symmetrical and is exposed uniformly on all sides to the fire. The finite element mesh used for the column quadrant is shown in Fig. 7.18(b). The planes of conduction symmetry are modeled as insulated surfaces (Q = 0). Because a steep temperature gradient is expected to exist near the column surface, a finer mesh size is used in this region.

The transient temperature distributions in the column were calculated using the program DOT with a time step of 0.025 hours and reforming the system equations every 4th time step. The same problem was also solved using the finite element program FIRES-T, which uses an algorithm involving heat flow equilibrium iteration [40].

Figure 7.19 compares the predicted temperature histories obtained from the DOT and FIRES-T analyses for certain reference elements: the corner element (64), the midside element (8) and the center element (1). The results of the two analyses are seen to be in close agreement for elements 1 and 8, but differ slightly for the corner element 64. The reason for this discrepancy between the two analyses for element 64 is not altogether clear. The major portion of the difference occurred in the corner node 81 (Fig. 7.18(b)) with nodes 71, 72 and 80 being much closer in agreement. Part of the difference may be due to the type of element used on the respective programs; DOT uses a variable 4- to 8-node isoparametric element with 2x2 Gauss point integration, whereas FIRES-T uses a quadrilateral element constructed from four linear triangles with static condensation of the central node assuming that no external heat flow occurs at that point. Additionally, the linear convection boundary condition is solved exactly in the DOT program by making modifications to both the appropriate conductivity matrix terms and heat flux vector terms whereas FIRES-T uses an approximate solution technique where the heat flux vector alone is modified.

TABLE 7.2: ASTM STANDARD E-119 FIRE CURVE DATA FOR A LONG DURATION MODERATELY INTENSE FIRE

Time, min.	Temperature, ^{o}F
0	68
5	1000
10	1300
15	1399
20	1462
25	1510
30	1550
35	1584
40	1613
45	1638
50	1661
55	1681
60	1700



(a) Column Cross-Section

(b) Finite Element Mesh of Column Quadrant

FIG. 7.18 NONLINEAR HEAT CONDUCTION IN A SQUARE COLUMN UNIFORMLY EXPOSED TO ASTM FIRE

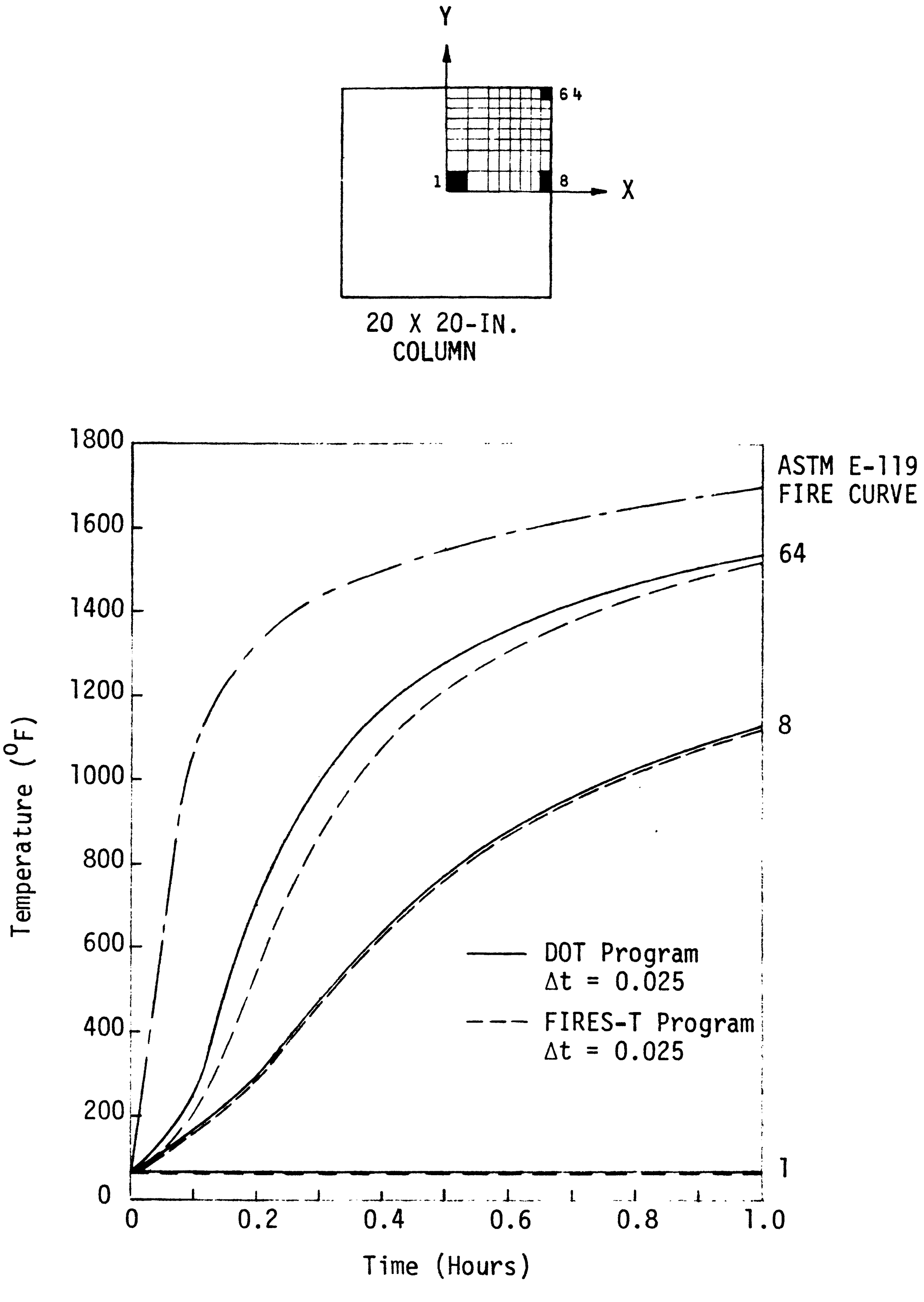


FIG. 7.19 COMPARISON OF PREDICTED ELEMENT TEMPERATURE HISTORIES FOR A SQUARE COLUMN UNIFORMLY EXPOSED TO ASTM E-119 FIRE

8. SUMMARY AND CONCLUSIONS

A finite element analysis technique for the solution of nonlinear transient heat conduction problems has been presented. Based on this theory, two finite element heat transfer programs have been developed. The program DOT ("Determination Of Temperatures") is capable of solving both linear and nonlinear heat transfer problems. Nonlinearities may be due to temperature-dependent material properties and, in particular, may be caused by nonlinear radiation boundary conditions. The program DETECT ("DEtermination of TEmperatures in ConsTruction") is a linear heat transfer analysis program developed with particular reference to mass concrete structures of arbitrary geometry constructed incrementally. Both programs can handle time-dependent boundary conditions and internal heat generation. A description of the required input data for the DOT and DETECT programs along with the Fortran IV listings is given in the Appendices of this dissertation.

A step-by-step solution technique has been described which is unconditionally stable for linear conduction problems and provides an efficient numerical solution of the heat flow equilibrium equations. The sample analyses presented in Section 7 show some applications of the method to different types of steady state and transient conduction problems and illustrate the major features of the programs.

For the solution of nonlinear heat transfer problems, it should be noted that the computational algorithm contained in the program DOT does not include an iterative scheme for heat flow equilibrium. Instead, a step-by-step solution procedure is used similar to the one outlined in Table 5.1 in which the effective thermal conductivity matrix $\underline{K}^*$ is reformed after every n user-specified time steps. For a particular time interval $\{t,t+\Delta t\}$, the nodal temperatures are estimated at the midpoint of this time step as $\underline{T}_t + \frac{1}{2}(T_t - T_{t-\Delta t})$. Temperature-dependent thermal properties are then computed based on this estimated midpoint temperature and the effective conductivity matrix $\underline{K}^*$ is updated. Because the thermal properties of most engineering materials vary gradually with temperature, this approximate procedure for treating the material nonlinearities was judged to be sufficiently accurate for a large class of problems and an iterative heat flow equilibrium scheme was not deemed necessary.

The capabilities of the DOT and DETECT programs may be expanded by including new finite element groups such as one dimensional beam elements and three dimensional isoparametric solid elements. Because the programs have been developed in a modular fashion, these new element groups can be added by developing a few new subroutines with only minor changes to the existing programs.

REFERENCES

1. Zienkiewicz, O. C. and Cheung, Y. K., "Finite Elements in the Solution of Field Problems," The Engineer, September 24, 1964.

2. Wilson, E. L., "A Digital Computer Program for the Steady-State Temperature Analysis of Plane or Axisymmetric Bodies,"Aerojet-General Corporation, Sacramento, California, Report No. TD-44, March 1965.

3. Wilson, E. L. and Nickell, R. E., "Application of the Finite Element Method to Heat Conduction Analysis," Nuclear Engineering and Design (Holland), Vol. 4, pp. 276-286, October 1966.

4. Carslaw, H. S. and Jaeger, J. C., "Conduction of Heat in Solids," 2nd ed., Oxford University Press, 1969.

5. Loretan, P. A., "Laplace Variational Method for Transient Multidimensional Temperature Distributions," Nuclear Engineering and Design, No. 11, pp. 27-40, 1970.

6. Goodman, T., "Application of Integral Methods to Transient Nonlinear Heat Transfer," Vol. 1, Academic Press, New York, pp. 51-122, 1964.

7. Trent, D. S. and Welty, J. R., "A Summary of Numerical Methods for Solving Transient Heat Conduction Problems," Bulletin No. 49, Engineering Experiment Station, Oregon State University, Corvallis, October 1974.

8. Ergatoudis, I., Irons, B. M. and Zienkiewicz, O. C., "Curved Isoparametric Quadrilateral Elements for Finite Element Analysis," International Journal of Solids and Structures, Vol. 4, pp. 31-42, 1968.

9. Biot, M. A., "Thermo-Elasticity and Irreversible Thermodynamics," J. Appl. Phys., Vol. 27, No. 3, pp. 240-253, March 1956.

10. Biot, M. A., "New Methods in Heat Flow Analysis with Applications to Flight Structures," Journal of the Aeronautical Sciences, Vol. 24, No. 12, pp. 857-873, December 1957.

11. Biot, M. A., "Further Developments of New Methods in Heat-Flow Analysis," Journal of the Aero/Space Sciences, Vol. 26, No. 6, pp. 367-381, June 1959.

12. Lardner, T. J., "Biot's Variational Principle in Heat Conduction," AIAA Journal, Vol. 1, No. 1, pp. 196-206, January 1963.

13. Gurtin, M. E., "Variational Principles for Linear Initial-Value Problems," Quarterly of Applied Mechanics, Vol. 22, No. 3, pp. 252-256, 1964.

REFERENCES (CON'T)

14. Gallagher, R. H. and Mallet, R., "Efficient Solution Processes for Finite Element Analysis of Transient Heat Conduction," ASME Paper No. 69-WA/HT-32, November 1969.

15. Rybicki, E. F. and Hopper, A. T., "Higher Order Finite Element Method for Transient Temperature Analysis of Inhomogenous Materials," ASME Paper No. 69-WA/HT-33, November 1969.

16. Aguirre-Ramirez, G. and Oden, J. T., "Finite-Element Technique Applied to Heat Conduction in Solids with Temperature Dependent Thermal Conductivity," ASME Paper No. 69-WA/HT-34, November 1969.

17. Richardson, P. D. and Shum, Y. M., "Use of Finite Element Methods in Solution of Transient Heat Conduction Problems," ASME Paper No. 69-WA/HT-36, November 1969.

18. Brocci, R. A., "Analysis of Axisymmetric Linear Heat Conduction Problems by Finite Element Method," ASME Paper No. 69-WA/HT-37, November 1969.

19. Emery, A. F. and Carson, W. W., "An Evaluation of the Use of the Finite Element Method in the Computation of Temperature," ASME Paper No. 69-WA/HT-37, November 1969.

20. Taylor, R. L., "*HEAT*, A Finite Element Computer Program for Heat Conduction Analysis" (Unpublished), University of California, Berkeley, May 1975.

21. Finlayson, B. A. and Scriven, L. E., "The Method of Weighted Residuals and its Relation to Certain Variational Principles for the Analysis of Transport Processes," Chemical Engineering Science, Vol. 20, pp. 395-404, 1965.

22. Wilson, E. L., "The Determination of Temperatures Within Mass Concrete Structures," Structural Engineering Laboratory Report No. 68-17, University of California, Berkeley, December 1968.

23. Farhoomand, I. and Wilson, E. L., "Non-Linear Heat Transfer Analysis of Axisymmetric Solids," SESM Report No. 71-6, University of California, Berkeley, April 1971.

24. Wilson, E. L., Bathe, K. J. and Peterson, F. E., "Finite Element Analysis of Linear and Nonlinear Heat Transfer," 2nd International Conference on Structural Mechanics in Reactor Technology, Berlin, Germany, Vol. V - Part L, September 1973.

25. Peterson, F. E. and Hui, H., "Sample Solutions and Theoretical Basis for 3D Transient Heat Transfer by Finite Element Procedure," Engineering/Analysis Corporation, Report No. 4.1.4.1, Berkeley, California, August 1971.

REFERENCES (CON'T)

26. Zienkiewicz, O. C., "The Finite Element Method in Engineering Science," McGraw-Hill, London, 1971.

27. Wilson, E. L., "SAP, A General Structural Analysis Program," University of California Structural Engineering Laboratory Report No. UC SESM 70-20, September 1970.

28. Wilson, E. L., "SOLID SAP, A Static Analysis Program for Three Dimensional Structures," University of California Structural Engineering Laboratory Report No. UC SESM 71-19, Sept. 1971.

29. Kreith, F., "Principles of Heat Transfer," Intext Press Inc., New York, 1973.

30. Bathe, K. J., Wilson, E. L. and Iding, R. H., "NONSAP, A Structural Analysis Program for Static and Dynamic Response of Nonlinear Systems," University of California Structural Engineering Laboratory Report No. UC SESM 74-3, February 1974.

31. Wilson, E. L., Bathe, K. J. and Doherty, W. P., "Direct Solution of Large Systems of Linear Equations," Computers and Structures, Vol. 4, No. 2, pp. 363-372, March 1974.

32. Peavy, B. A., "Steady-State Heat Conduction in an Exposed Exterior Column of Rectangular Cross Section," J. Res. National Bureau of Standards, 69c, p. 145, 1965.

33. Irving and Mullineax, "Mathematics in Physics and Engineering," Academic Press Inc., New York, 1969.

34. Carnahan, Luther and Wilkes, "Applied Numerical Methods," John Wiley and Sons, New York, 1969.

35. Olson and Schultz, "Temperatures in Solids During Heating or Cooling," Ind. Eng. Chem., pp. 874-877, 1942.

36. Clough, R. W., "The Stress Distribution of Norfork Dam," Institute of Engineering Research Report Series 100, Issue 19, University of California, Berkeley, August 1962.

37. Pirtz, D., Klein, A. and Trescony, L. J., "Thermal Properties of Lean Mass Concretes for Oroville Dam," Structural Engineering Laboratory Report No. 63-3, University of California, Berkeley, June 1963.

38. American Society for Testing and Materials, "Standard Methods of Fire Tests of Building Construction and Materials E119-73," Book of ASTM Standards, Part 14, Philadelphia, p. 462, November 1970.

REFERENCES (CON'T)

39. Bizri, H., "Structural Capacity of Reinforced Concrete Columns Subjected to Fire Induced Thermal Gradients", Report No. UC SESM 73-1, Department of Civil Engineering, University of California, Berkeley, 1973.

40. Becker, J., Bizri, H. and Bresler, B., "FIRES-T - A Computer Program for the Fire Response of Structures - Thermal", Report No. UCB FRG 74-1, Fire Research Group, Department of Civil Engineering, University of California, Berkeley, 1974.

APPENDIX A1

DOT USER'S MANUAL

PROGRAM IDENTIFICATION

DOT: A Nonlinear Determination of Temperatures Program for Analysis of Two Dimensional Planar or Axisymmetric Structures. Version I, Febuary 1976.

Developed by: R. M. Polivka and E. L. Wilson
University of California, Berkeley

CONTENTS

A1.1 PROGRAM DESCRIPTION

The computer program DOT performs the steady state or transient heat transfer analysis of two dimensional planar or axisymmetric structures subjected to surface heat flux, convective and radiative heat transfer, internal heat generation, temperature boundary conditions or artificial cooling by means of embedded pipes. The program is capable of solving both linear and nonlinear analysis problems. The nonlinearities may be due to either temperature dependent material properties of nonlinear boundary conditions.

The program is completely general, and is applicable to structures of arbitrary geometry composed on several different materials. The present version contains the following element types:

1. Two dimensional 4- to 8-node planar or axisymmetric solid elements.
2. Convection and radiation 2-node boundary elements.
3. Cooling pipe elements.

A.1.2 PROGRAM CAPACITY

The computer program DOT uses variable dimensioning in order to make optimum use of the available high speed storage. The element data for each group of elements is stored in a block and transferred to disc storage. The required core storage remaining is dynamically allocated into a single array A in blank common.

The program capacity can be controlled by the user through the two fortran statements in the main program of DOT:

COMMON (n)

MTOT = n

The total memory n required can be changed depending on the size of the problem to be solved. The minimum value of n needed is given by

$$n = M_{max} + 2*NPTM*NBCF + NBCF + 4*NNBC + 6*NUMNP + NWK + 1$$

where

M_{max} = Maximum storage required for element group data

NPTM = Maximum number of time points used to describe any of the boundary condition functions

NBCF = Number of boundary condition functions

NNBC = Number of nodal point boundary conditions

NUMNP = Number of nodes

NWK = Working storage required for conductivity matrix. Approximate by MB*NUMNP where MB = maximum bandwidth.

M_{max} is the maximum value of M for each of the three element types used in the analysis:

(a) Type 1 - Planar or axisymmetric elements

$$M_1 = NEL1*(4*MXNODS - 2) + NUMAT*(5*MAXTP + 4)$$

NEL1 = Number of elements in group 1

NUMAT = Number of material types

MXNODS = Maximum number of nodes defining any element

MAXTP = Maximum number of temperatures defining a given material

(b) Type 2 - Convection and radiation surface elements

M_2 = 6*NEL2 + 7*NUMAT + 2

NEL2 = Number of elements in group 2

NUMAT = Number of material types

(c) Type 3 - Cooling pipe elements

M_3 = 6*NEL3

NEL3 = Number of elements in group 3

If the value of n is set too small, an error message is printed which gives the amount by which the storage was exceeded and the program execution is terminated.

A1.3 PROGRAM INPUT DATA

The following sections describe the necessary sequence of cards which define a given structure to be analyzed.

I. PROBLEM INITIATION AND TITLE (A5,3X,18A4) - One card

Columns	Variable	Description
1 - 5	MODE	Problem initiation flag. Punch the word START.
6 - 8	--	Blank
9 - 80	HED	Problem title for labeling output.

II. MASTER CONTROL CARD (4I5) - One card

Columns	Variable	Description
1 - 5	NUMNP	Total number of nodal points in structure.
6 - 10	NEG	Number of element groups. See Note 1.
11 - 15	NUMEST	Estimated maximum number of high speed storage locations M_{max} required to store each set of element group data. Specify (a) Zero or blank: defaults to 4000.
16 - 20	MODEX	Execution mode. Specify (a) Zero or blank: data check only. (b) 1: execution.

NOTE

(1) An element group is a series of elements of a particular type (e.g. 2/D axisymmetric, 2/D planar, convection elements etc.). Elements of the same type may also be split into more than one group.

III. NODAL POINT COORDINATES (I5,5X,2F10.0,I5,I1)

As many cards as needed to generate NUMNP nodal points.

Columns	Variable	Description
1 - 5	N	Node Number. See Note 1.
6 - 10	--	Blank
11 - 20	X(N)	X coordinate.
21 - 30	Y(N)	Y coordinate.
31 - 35	KN	Node number difference between successive generated nodes (given on first card in a sequence). Specify (a) Zero or blank: No generation. See Note 2.
36	JPR	Print suppression flag (on card for node 1 only). Specify (a) Zero or blank: No suppression. (b) 1: Suppress ordered list of node coordinates. See Note 3.

NOTE

(1) Node cards need not be in numerical order. Eventually, however, all nodes from 1 to the total number of nodes (NUMNP) must be identified.

(2) The mesh generation parameter KN must appear on the first card of a series of nodal points to be generated. The intermediate nodes to be generated between nodes N1 and N2 will be located at equal intervals along the straight line joining the two nodes. KN is the increment to be added to the previous node number. The node difference N2-N1 must be evenly divisible by KN.

(3) JPR is used to eliminate the second printing of the ordered node coordinates. The JPR character is entered on the card for node 1 only.

IV. SOLUTION TIME AND TEMPERATURE CONTROL CARD (2I5,3F10.0,3I5)

Columns	Variable	Description
1 - 5	KST	Code for steady state or transient analysis. (a) -1: Steady state analysis. (b) Zero or blank: Transient analysis.
6 - 10	NDT	Number of solution time steps. Specify (a) Zero or blank: defaults to 1.
11 - 20	DT	Time step increment.
21 - 30	TSTART	Time at solution start. See Note 1.
31 - 40	TAMB	Ambient temperature. See Note 2.
41 - 45	NPRINT	Time interval for printout of nodal temperatures, expressed as a multiple of the integration time step. Specify (a) Zero or blank: defaults to 1.
46 - 50	NTSREF	Number of time steps between reforming effective conductivity matrix. Specify (a) Zero or blank: Conductivity matrix never reformed. (b) n: Conductivity matrix reformed every n steps.
51 - 55	KP	Time interval for punched output of nodal temperatures, expressed as a multiple of the integration time step. Specify (a) Zero or blank: No punched output. (b) n: Punched output every n steps. See Note 3.

NOTE

(1) The time at solution start (TSTART) is an input convenience for restart jobs. In a restart job, TSTART is the final time to which a previous job was run and the nodal temperatures saved for use in supplying initial conditions for a new job. The time counter is incremented from TSTART, so none of the boundary condition functions in Section V need be revised.

(2) The body is assumed to be at uniform ambient temperature (TAMB) prior to the application of any heat flow boundary conditions; i.e. at t = 0 all nodes are at temperature (TAMB).

(3) The punched output time interval KP controls punching of nodal temperatures. For KP = n, the temperatures at the end of every n steps are punched in exactly the same format as the initial conditions in Section VII. In addition, the nodal point coordinate data is also punched for possible use with a mesh plotting program.

For the case of KP = NDT, the nodal temperatures at the end of the analysis will be saved on punched cards, which in turn can be read in as initial conditions in a restart job. By setting TSTART = NDT*DT, a new analysis run can now be made.

V. BOUNDARY CONDITION FUNCTIONS

A. CONTROL INFORMATION (3I5) - One card

Columns	Variable	Description
1 - 5	NBCF	Number of boundary condition functions. See Note 1.
6 - 10	NPTM	Maximum number of points used to describe any one of the functions. See Note 2.
11 - 15	NNBC	Number of nodal point heat flow or temperature boundary conditions. See Note 3.

NOTE

(1) NBCF determines the number of card sets to be read in Section V.B.

(2) NPTM is the maximum number of [f(t),t] pairs used to define any one of the NBCF functions. At least two points are required to input any one function, and no function may be input with more than NPTM points.

(3) NNBC determines the number of cards to be read or generated in Section VI.

B. TIME FUNCTION DATA - NBCF sets of cards

Each set consists of a control card followed by as many cards as needed to define the function.

1. CONTROL CARD (2I5) - First card of set.

Columns	Variable	Description
1 - 5	NC	Function number. (GE.1 and LE.NBCF) See Note 1.
6 - 10	NPTS(NC)	Number of time points used to describe this function. (GE.2 and LE.NPTM)

2. [f(t),t] DATA (8F10.0) - Remaining card(s) of set.

As many cards as needed to define NPTS(NC) pairs of points [TFN(NC,I),FN(NC,I)], four pairs per card. See Note 2.

Columns	Variable	Description
1 - 10	TFN(NC,1)	Time at point 1 : t_1
11 - 20	FN (NC,1)	Function value at point 1 : $f(t_1)$
21 - 30	TFN(NC,2)	Time at point 2 : t_2
31 - 40	FN (NC,2)	Function value at point 2 : $f(t_2)$
41 - 50	TFN(NC,3)	Point 3
51 - 60	FN (NC,3)	
61 - 70	TFN(NC,4)	Point 4
71 - 80	FN (NC,4)	

NEXT CARD(S) - If required

1 - 10	TFN(NC,5)	Time at point 5 : t_5
11 - 20	FN (NC,5)	Function value at point 5 : $f(t_5)$
. . .	. . .	. . .

NOTE

(1) Time functions need not be input in order of increasing function number NC.

(2) The function tables input in this section are used to prescribe time dependent boundary conditions such as environmental or nodal point temperatures and nodal point heat flows. The functions may also be used to describe the rate of internal heat generation and the variation of convection coefficient H with temperature.

Time values at successive points must increase in magnitude. Values of the functions at times (temperatures) other than TFN(NC,I) are calculated within the program using linear interpolation.

The first time point TFN(NC,1) must be less than or equal to the time at solution start TSTART and the final time point TFN(NC,NPTS(NC)) must be greater than or equal to the time at the end of solution NDT*DT.

VI. NODAL POINT BOUNDARY CONDITIONS (4I5,F10.0,I5) - NNBC cards

Columns	Variable	Description
1 - 5	N	Boundary condition number. See Note 1.
6 - 10	NOD(N)	Global node number. (GE.1 and LE.NUMNP)
11 - 15	KODE(N)	Boundary condition type. Specify (a) Zero or blank : Externally supplied heat flux Q. (b) 1: Prescribed temperature T.
16 - 20	NFN(N)	Function number. (GE.0 and LE.NBCF) See Note 2.
21 - 30	TQ(N)	Boundary value amplitude. Specify (a) Heat flux (KODE(N) = 0). (b) Temperature (KODE(N) = 1). See Note 3.
31 - 35	KN	Node number difference between successive generated nodes (on first card in sequence). See Note 4.

NOTE

(1) All nodal points not specified in this section are assumed to have externally supplied heat flux of zero for all values of time.

(2) A function number equal to zero or blank means that the prescribed boundary condition is applied at time zero and remains constant for all time greater than zero (step function). The functions assigned in this section must have been defined previously in Section V. A given function can be used to describe any number of boundary conditions.

(3) (a) For the case of flux or temperature boundary conditions which are not step functions, TQ(N) represents a function multiplier used to scale NFN(N) values for all time t.

(b) For the case of flux or temperature boundary conditions applied as a step function from time zero, TQ(N) represents the actual value of prescribed flux or temperature.

(4) In the printout of nodal point boundary conditions, all generated node data are prefixed by an asterisk. For use of the generation parameter KN, see Note 2, Section III.

VII. INITIAL CONDITIONS

This set of cards consists of a control card followed by as many cards as needed to define the initial nodal temperatures not equal to the specified ambient temperature (TAMB).

A. CONTROL CARD (I5) - One card

Columns	Variable	Description
1 - 5	ICON	Initial condition flag. Specify (a) Zero or blank: Temperatures of all nodes set automatically to the specified ambient temperature. (b) n > 0: Nodes having user supplied initial conditions are read from data cards in Section VII.B. See Note 1.

B. INITIAL TEMPERATURES (4(I5,5X,E10.0)) - n temperatures

As many cards as needed to specify n nodal temperatures (four values per card) which are not equal to the ambient temperature. Omit if the initial condition flag ICON equals zero or blank. See Note 2.

Columns	Variable	Description
1 - 5	ND	Node number.
11 - 20	T(ND)	Temperature.
. . .	. . .	. . .

NOTE

(1) The initial condition flag ICON is used to control card reading in Section VII.B. For ICON equal to zero, the initial temperatures are set equal to the specified ambient temperature TAMB and input should then resume beginning in Section VIII.

(2) For a restart job where the initial conditions are supplied from a previous heat transfer solution performed using DOT, the variable ICON must equal NUMNP and the initial temperatures then read in Section VII.B will be the punched deck obtained from the first run using KP equal to NDT in Section IV.

VIII. ELEMENT SPECIFICATION

Elements must be divided into "groups". Input as many blocks of data in the following sections as there are element groups (NEG). An element group is a series of elements of a particular type, and elements of the same type may also be divided into more than one group.

Element groups may be input in any order. The elements in any group must be numbered sequentially starting from the number of the first element as specified on the element group control card.

The following types of element groups may be used:

Type 1 - Two Dimensional Finite Elements

These are 4- to 8-node isoparametric elements which must be input in the global X-Y plane. When the element is used to represent an axisymmetric solid, the global Y-axis is the axis of revolution.

Type 2 - Convection and Radiation Surface Elements

These are 2-node planar or axisymmetric solid boundary elements which must be input in the global X-Y plane.

Type 3 - Cooling Pipe Elements

These are 1-node elements which are placed at existing nodal points in the finite element mesh. The axis of the cooling pipe is the Z-axis.

TYPE 1 - TWO DIMENSIONAL FINITE ELEMENTS

A. CONTROL INFORMATION (9I5) - One card

Columns	Variable	Description
1 - 5	NGR	Element group indicator. Punch the number "1".
6 - 10	NEL1	Number of elements in this group.
11 - 15	MFST	Element number of first element in group. (a) Zero or blank: defaults to 1. See Note 1.
16 - 20	ITYP2D	Element type code. Specify (a) Zero or blank: axisymmetric (b) 1: planar
21 - 25	MXNODS	Maximum number of nodes used to describe any one element. Specify (a) Zero or blank: defaults to 4. (GE.4 and LE.8)
26 - 30	NINT	Numerical integration order to be used in Gaussian quadrature. Specify (a) Zero or blank: defaults to 2. (GE.2 and LE.4) See Note 2.
31 - 35	NUMAT	Number of material property sets. Specify (a) Zero or blank: defaults to 1.
36 - 40	MAXTP	Maximum number of temperature points in material table. Specify (a) Zero or blank: defaults to 1. (GT.0)
41 - 45	IHFLG	Internal heat generation flag. Specify (a) Zero or blank: No internal heat generation. (b) 1: Internal heat generation exists.

NOTE

(1) Element numbers need not begin with number 1 if MFST is specified.

(2) For rectangular elements, an integration order of 2 is sufficient. If the element is distorted, a higher integration order need be used.

B. MATERIAL PROPERTY INFORMATION - NUMAT sets of cards.

1. MATERIAL IDENTIFICATION CARD (I5,5X,F10.0,2I5,F10.0) - One card

Columns	Variable	Description
1 - 5	M	Material identification number. (GE.1 and LE.NUMAT)
6 - 10	--	Blank
11 - 20	DENS(M)	Density of material. Leave blank for steady-state problems.
21 - 25	NTC(M)	Number of temperature points describing the material. Specify (a) Zero or blank: defaults to 1.
26 - 30	NFHG(M)	Function number describing the rate of internal heat generation. See Note 1.
31 - 40	HGSC(M)	Internal heat generation function multiplier. See Note 1.

2. MATERIAL PROPERTY CARD(S) (5F10.0) - Remaining NTC(M) cards of set. See Note 2.

Columns	Variable	Description
1 - 10	TPROP(1)	Temperature T_1.
11 - 20	TPROP(2)	Conductivity k_{11} at temperature T_1.
21 - 30	TPROP(3)	Conductivity k_{22} at temperature T_1.
31 - 40	TPROP(4)	Conductivity k_{12} at temperature T_1.
41 - 50	TPROP(5)	Specific heat c at temperature T_1.

NOTE

(1) If no internal heat generation exists, i.e. if IHFLG equals zero, leave columns 26-40 blank. For the case where internal heat generation is considered, the functions NFHG assigned in this section must have been defined previously in Section V. See Note 2, Section VI.

(2) Consistent units must be used for specification of material properties.

C. ELEMENT DATA (12I5)

As many cards as needed to generate NEL1 elements.

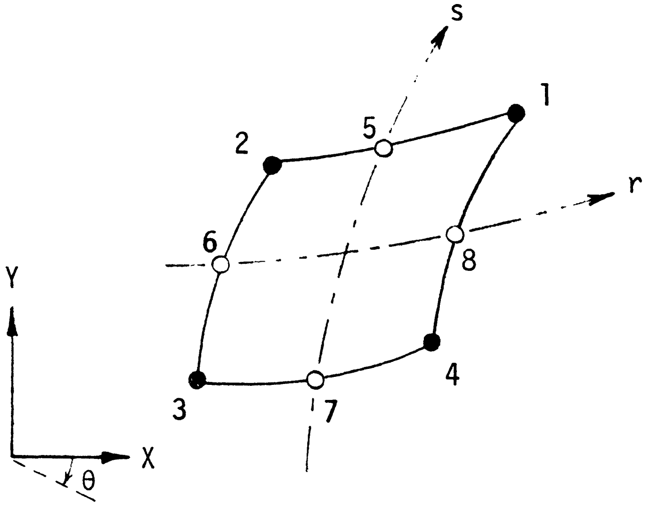
Columns	Variable	Description
1 - 5	M	Element number. See Note 1.
6 - 10	NOD(1)	Global node number of element node 1.
11 - 15	NOD(2)	Global node number of element node 2.
16 - 20	NOD(3)	Global node number of element node 3.
21 - 25	NOD(4)	Global node number of element node 4.
26 - 30	NOD(5)	Global node number of element node 5.
31 - 35	NOD(6)	Global node number of element node 6.
36 - 40	NOD(7)	Global node number of element node 7.
41 - 45	NOD(8)	Global node number of element node 8. See Note 2.
46 - 50	MTYP	Material identification number. Specify (a) Zero or blank: defaults to 1. (GE.1 and LE.NUMAT)
51 - 55	IEL	Number of nodes used to describe element. (a) Zero or blank: defaults to MXNODS.
56 - 60	KG	Node number increment for element generation (given on first card in a sequence). (a) Zero or blank: defaults to 1. See Note 3.

NOTE

(1) All elements must be input in increasing numerical order, starting with element number MFST. Cards for the first and last element must be included.

(2) The number of nodes in element M is defined by IEL. For elements containing less than eight nodes (IEL.LT.8), input a zero or blank in NOD(I) for the particular node locations not used. As an example, for a 6-node element (IEL.EQ.6) with nodes 5 and 7 not used, the element node number array would be NOD(I) = [X X X X 0 X 0 X] where the nonzero entries (X) are the global node numbers of the 6 nodes.

Figure VIII-1 defines the mapping between the node numbering in the local r-s coordinate system and the global X-Y coordinate system. Nodes 1 through 4 are located at the four corners in a counterclockwise sense, and nodes 5 through 8 are located at the midsides of the element.



Node	r	s
1	+1	+1
2	-1	+1
3	-1	-1
4	+1	-1
5	0	+1
6	-1	0
7	0	-1
8	+1	0

FIG. VIII-1 TWO DIMENSIONAL FINITE ELEMENT NODE NUMBER INPUT SEQUENCE

(3) The node generation parameter KG must appear on the first element card of a sequence, and is used to compute the node numbers for a group of missing elements.

If data for elements [M+1,M+2,... M+J] are omitted, these "J" missing elements are generated using the same values for material number (MTYP) and number of nodes (IEL) as given on the preceding card for element M, and the node numbers for the successive "J" elements are incremented by the value KG given on the M-th element card. Only the nonzero node numbers appearing on the M-th element card are incremented when generating missing element data.

In the printout of the element data, generated elements are prefixed by an asterisk.

TYPE 2 - CONVECTION AND RADIATION ELEMENTS

A. CONTROL INFORMATION (5I5) - One card

Columns	Variable	Description
1 - 5	NGR	Element group indicator. Punch the number "2".
6 - 10	NEL2	Number of elements in this group.
11 - 15	MFST	Element number of first element in group. (a) Zero or blank: defaults to 1. See Note 1.
16 - 20	ITYP	Element type code. Specify (a) Zero or blank : axisymmetric solid boundary element. (b) 1: Planar solid boundary element.
21 - 25	NUMAT	Number of material property sets. Specify (a) Zero or blank: defaults to 1.

B. RADIATION CONSTANTS CARD (2E10.0) - One card

Columns	Variable	Description
1 - 10	SB	Stefan-Boltzmann constant. See Note 2.
11 - 20	TSHIFT	Shift for absolute temperature. See Note 3.

NOTE

(1) Element numbers need not begin with number 1 if MFST is specified.

(2) The Stefan-Boltzmann constant for radiation has the value of 0.1714×10^{-8} BTU/hr-ft^{2}-^{o}R^{4} in the engineering system of units. Care must be taken to use consistent units for all input data.

(3) In radiation phenomena, the temperature is measured above absolute zero (-460 F or -273 C) and is expressed in degrees Rankine (R). Depending on the units used for temperature, TSHIFT will equal either 460 or 273.

C. MATERIAL PROPERTY INFORMATION - NUMAT sets of cards

1. MATERIAL IDENTIFICATION CARD (3I5) - First card of set.

Columns	Variable	Description
1 - 5	M	Material identification number. (GE.1 and LE.NUMAT)
6 - 10	NFH	Function number describing the variation of convection coefficient H with temperature. See Note 1.
11 - 15	NFTE	Function number describing the environmental temperature variation with time. See Note 2.

2. MATERIAL PROPERTY CARD(S) (5F10.0) - Second card of set.

Columns	Variable	Description
1 - 10	PROP(1)	Convection coefficient function multiplier. See Note 1.
11 - 20	PROP(2)	Environmental temperature function multiplier. See Note 2.
21 - 30	PROP(3)	Radiation shape factor (view factor). (a) Zero or blank: radiation effects are ignored.
31 - 40	PROP(4)	Surface emissivity.
41 - 50	PROP(5)	Environment emissivity.

NOTE

(1) The variation of convection coefficient H with temperature is computed using H(T) = PROP(1)*NFH where the function number NFH has been defined previously in Section V. A function number equal to zero or blank will be assumed to have the temperature independent value of unity.

(2) The variation of environmental temperature with time is computed using $T_e(t)$ = PROP(2)*NFTE where the function number NFTE has been defined previously in Section V. If the environmental temperature does not change with time, input NFTE equal to zero or blank and PROP(2) = T_o, constant for all time greater than zero (step function).

D. ELEMENT DATA (5I5)

As many cards as needed to generate NEL2 elements.

Columns	Variable	Description
1 - 5	M	Surface number. See Note 1.
6 - 10	NOD(1)	Global node number of element node I.
11 - 15	NOD(2)	Global node number of element node J. See Note 2.
16 - 20	MTYP	Material identification number. Specify (a) Zero or blank: defaults to 1. (GE.1 and LE.NUMAT)
21 - 25	KG	Node number increment for element generation (given on first card in a sequence). (a) Zero or blank: defaults to 1. See Note 3.

NOTE

(1) All elements must be input in ascending numerical order, starting with element number MFST. Cards for the first and last element must be included.

(2) Convection and radiation boundary elements are 2-node elements on the surface of a planar or axisymmetric solid, as shown in Figure VIII-2.

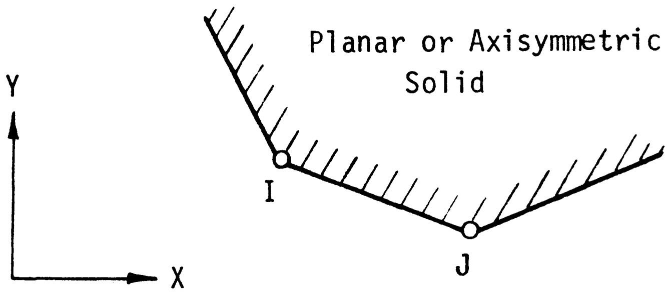


FIG. VIII-2 CONVECTION AND RADIATION SURFACE ELEMENTS

(3) The node generation parameter KG must be given on the first element card prior to a group of missing elements. All generated elements will have the same material identification number (MTYP) as that given on the first element card in the sequence. In the printout of the element data, generated elements are prefixed by an asterisk.

TYPE 3 - COOLING PIPE ELEMENTS

A. CONTROL INFORMATION (3I5) - One card

Columns	Variable	Description
1 - 5	NGR	Element group indicator. Punch the number "3".
6 - 10	NEL3	Number of elements in this group.
11 - 15	MFST	Element number of first element in group. (a) Zero or blank: defaults to 1. See Note 1.

B. ELEMENT DATA (2I5,F10.0,2I5,3F10.0)

As many cards as needed to generate NEL3 elements.

Columns	Variable	Description
1 - 5	M	Element number. See Note 1.
6 - 10	NOD	Global node number of cooling pipe location.
11 - 20	HP	Empirical constant. See Note 2.
21 - 25	NFC	Function number describing the cooling water temperature variation with time. See Note 3.
26 - 30	KG	Node number increment for element generation (given on first card in a sequence). See Note 4.
31 - 40	TWTR	Cooling water temperature function multiplier. See Note 3.
41 - 50	TBTH	Time at initiation of cooling.
51 - 60	TDTH	Time when cooling is to be stopped.

NOTE

(1) Element numbers need not begin with number 1 if MFST is specified. All elements must be input in increasing numerical order starting with element MFST. The first and last element cards must be included.

(2) The rate at which heat is removed from the solid by a cooling pipe is given by:

$$q = HP(T_w - T_o)$$

where

T_w = temperature of the cooling water.

T_o = apparent temperature on the outer surface of the pipe at node NOD in the finite element solution.

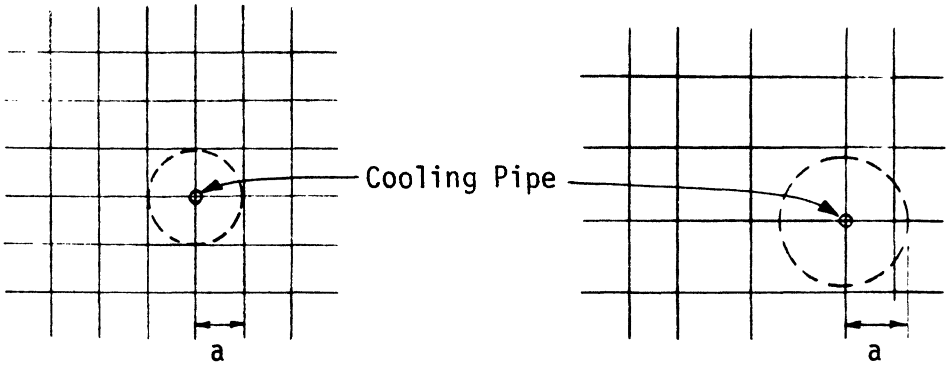
$$HP = \frac{2\pi k}{\ln(\frac{a}{R})-2}$$, an empirical constant.

where

k = average conductivity of finite elements adjacent to the cooling pipe.

R = radius of cooling pipe.

a = weighted distance to adjacent node. (See Fig. VIII-3)



(i) Regular Mesh

(ii) Irregular Mesh

FIG. VIII-3 DETERMINATION OF 'a' IN EQUATION FOR HP

(3) If the cooling water temperature T_w is time-dependent, then a nonzero function number NFC must be specified. If the cooling water temperature T_w does not change with time, then input NFC equal to zero or blank and TWTR = T_o, constant for all time greater than zero (step function).

(4) The node generation parameter KG must appear on the first element card of a sequence, and is used to compute node numbers for missing elements. All data for generated elements is taken to be the same as that given on the first element card in the sequence. In the printout of the element data, all generated elements are prefixed by an asterisk.

IX. NEW PROBLEM DATA

A completely new problem may now be solved by starting with Section I. Any number of heat transfer problems may be solved within a single computer run.

X. TERMINATION CARD (A4) - One card

Columns	Variable	Description
1 - 4	MODE	Problem(s) termination flag. Punch the word STOP.

APPENDIX A2

DOT FORTRAN IV LISTING

```
      PROGRAM DOT (INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
     1                TAPE1,TAPE2,TAPE3=PUNCH)
C
C
C     ******************************************************************
C
C     DOT - A NONLINEAR DETERMINATION OF TEMPERATURES PROGRAM FOR
C           TWO DIMENSIONAL PLANAR OR AXISYMMETRIC STRUCTURES
C
C     DEVELOPED BY - R.M. POLIVKA AND E.L. WILSON , FEBUARY, 1976
C
C     ******************************************************************
C
C
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      COMMON /WORK  / WORK(200)
      COMMON  A(10000)
C
      MTOT   = 10000
  200 MAXEST = 0
C
C     ===========
C     INPUT PHASE
C     ===========
C
C     PROGRAM MASTER CONTROL DATA
C
      CALL DOTI
C
C     INPUT ELEMENT INFORMATION
C
      CALL ELCAL
C
C     ==============
C     SOLUTION PHASE
C     ==============
C
C     ******************************************************************
C
C     B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C           ARRAY  -----------DESCRIPTION-----------   DIMENSION
C     N1    TFN    TIME VALUES AT POINTS               NPTM*NBCF
C     N2    FN     FUNCTION VALUES AT POINTS           NPTM*NBCF
C     N3    NPTS   NUMBER OF FUNCTION INPUT POINTS     NBCF
C     N4    NOD    NODES HAVING PRESCRIBED BC          NNBC
C     N5    KODE   BOUNDARY CONDITION TYPES            NNBC
C     N6    NFN    BC FUNCTION NUMBERS                 NNBC
C     N7    TO     BOUNDARY VALUE AMPLITUDES           NNBC
C     N8    T      NODAL TEMPERATURES AT (T)           NUMNP
C     N9    MAXA   ADDRESSES OF XK DIAGONAL ELEMENTS   NUMNP+1
C     N10   XK     EFFECTIVE CONDUCTIVITY MATRIX       NWK
C     N11   Q      EXTERNALLY SUPPLIED HEAT FLUXES     NUMNP
C     N12   C      EFFECTIVE HEAT CAPACITY VECTOR      NUMNP
C     N13   E      THERMAL RESISTANCE VECTOR           NUMNP
C     N14   TT     NODAL TEMPERATURES AT (T-DT)        NUMNP
C
C     ******************************************************************
C
      CALL ADRSK (A(N11),A(N12),NUMNP,NWK,MB)
C
C     PUNCH NODAL POINT COORDINATE DATA
C
      IF(KP.EQ.0) GO TO 5
      CALL FNODE (A(N1),A(N2),NUMNP)
C
C     SHIFT STORAGE TO ELIMINATE NODAL COORDINATE DATA
C
    5 I = 1 + MAXEST
      N12M = N12 - 1
      DO 10 J=N3,N12M
      A(I) = A(J)
   10 I = I + 1
C
      N1  =  1   +  MAXEST
      N2  =  N1  +  NPTM*NBCF
      N3  =  N2  +  NPTM*NBCF
      N4  =  N3  +  NBCF
      N5  =  N4  +  NNBC
      N6  =  N5  +  NNBC
      N7  =  N6  +  NNBC
      N8  =  N7  +  NNBC
      N9  =  N8  +  NUMNP
      N10 =  N9  +  NUMNP + 1
      N11 =  N10 +  NWK
      N12 =  N11 +  NUMNP
      N13 =  N12 +  NUMNP
      N14 =  N13 +  NUMNP
      IF(KST.EQ.-1) N14 = N12
      N15 =  N14 +  NUMNP
      IF(KBC.EQ.0) N14 = N15
      IF(N15.GT.MTOT) CALL ERROR (N15-MTOT)
C
      IF(MODEX.EQ.0) GO TO 200
C
C     INITIALIZE CONDUCTIVITY MATRIX (XK) AND HEAT FLOW VECTOR (Q)
C
      N12M = N12 - 1
      DO 15 I=N10,N12M
   15 A(I) = 0.0
C
C     INITIALIZE TEMPERATURE VECTOR TT(0) = T(0)
C
      IF(KBC.EQ.0) GO TO 22
      DO 20 I=1,NUMNP
      IT  = N8  + I - 1
      ITT = N14 + I - 1
   20 A(ITT) = A(IT)
C
C     INITIALIZE THE TIME STEP COUNTER
C
   22 KSTEP = 0
      TIME  = TSTART
C
C     INITIALIZE EFFECTIVE HEAT CAPACITY VECTOR (C)
C
      IF(KST.EQ.-1) GO TO 30
      N13M = N13 - 1
      DO 25 I=N12,N13M
   25 A(I) = 0.0
C
C     ASSEMBLE THE EFFECTIVE SYSTEM CONDUCTIVITY MATRIX (K*)
C     BASED ON MATERIAL PROPERTIES AT TEMPERATURE T(0), AND
C     MODIFY FOR LINEAR CONVECTION AND COOLING PIPE BC
C
   30 CALL ASSEMK
C
      IF(NNBC.EQ.0) GO TO 35
C
C     MODIFY CONDUCTIVITY MATRIX FOR TEMPERATURE BOUNDARY CONDITIONS
C
      CALL MODKTB (A(N4),A(N5),A(N9),A(N10),NNBC)
C
   35 IF(KST.EQ.-1) GO TO 40
C
```

```
DOT  141  C     FORM THE INITIAL RESISTANCE VECTOR AT TIME ZERO, E(0)
DOT  142  C                        E(0) = [C*][T(0)]
DOT  143  C
DOT  144        CALL FORME (A(N8),A(N13),A(N12),NJMNP)
DOT  145  C
DOT  146  C     TRIANGULARIZE THE EFFECTIVE CONDUCTIVITY MATRIX, (K*)
DOT  147  C
DOT  148     40 KTR = 0
DOT  149        CALL COLSOL (A(N10),A(N11),A(N9),NUMNP,MB,NWK,KTR)
DOT  150  C
DOT  151  C     ==================
DOT  152  C     TIME MARCHING LOOP
DOT  153  C     ==================
DOT  154  C
DOT  155    10C KSTEP = KSTEP + 1
DOT  156        TIME  = TIME  + DT
DOT  157  C
DOT  158  C     CHECK TO SEE IF SYSTEM EQUATIONS TO BE REFORMED THIS TIME STEP
DOT  159  C
DOT  160        IF(KSTEP.EQ.1) GO TO 70
DOT  161        K = MOD(KSTEP,NTSREF)
DOT  162        IF(K.NE.0) GO TO 70
DOT  163  C
DOT  164  C     REFORM THE SYSTEM EQUATIONS
DOT  165  C
DOT  166        DO 45 I=N10,N12M
DOT  167     45 A(I) = 0.0
DOT  168        IF(KST.EQ.-1) GO TO 55
DOT  169        DO 50 I=N12,N13M
DOT  170     50 A(I) = 0.0
DOT  171  C
DOT  172     55 CALL ASSEMK
DOT  173  C
DOT  174        IF(NNBC.EQ.0) GO TO 60
DOT  175  C
DOT  176        CALL MODKTB (A(N4),A(N5),A(N9),A(N10),NNBC)
DOT  177  C
DOT  178     60 IF(KST.EQ.-1) GO TO 65
DOT  179  C
DOT  180        CALL FORME (A(N8),A(N13),A(N12),NUMNP)
DOT  181  C
DOT  182     65 KTR = 0
DOT  183        CALL COLSOL (A(N10),A(N11),A(N9),NUMNP,MB,NWK,KTR)
DOT  184  C
DOT  185  C     DEFINE EXTERNALLY SUPPLIED HEAT FLUX VECTOR (Q) FOR THIS TIME STEP
DOT  186  C
DOT  187     70 IF(NNBC.EQ.0) GO TO 75
DOT  188  C
DOT  189  C     PRESCRIBED (Q)
DOT  190  C
DOT  191        CALL FORMQP (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),
DOT  192       1             A(N11),NPTM)
DOT  193  C
DOT  194  C     CONVECTION, RADIATION AND INTERNAL HEAT GENERATION (Q)
DOT  195  C
DOT  196     75 CALL FORMQC
DOT  197  C
DOT  198        IF(KST.EQ.-1) GO TO 80
DOT  199  C
DOT  200  C     COMPUTE EFFECTIVE HEAT FLUX VECTOR
DOT  201  C       Q*(TIME) = Q(TIME) + E(TIME-1)
DOT  202  C
DOT  203        CALL QEFF (A(N11),A(N13),NUMNP)
DOT  204  C
DOT  205  C     UPDATE (TT) VECTOR
DOT  206  C
DOT  207     80 IF(KBC.EQ.0) GO TO 84
DOT  208        DO 82 I=1,NUMNP
DOT  209        IT  = N8  + I - 1
DOT  210        ITT = N14 + I - 1
```

```
DOT  211     82 A(ITT) = A(IT)
DOT  212  C
DOT  213  C     SOLVE THE HEAT EQUILIBRIUM EQUATIONS FOR THE SYSTEM TEMPERATURES
DOT  214  C               T(TIME) = [K*(TRI)][Q*(TIME)]
DOT  215  C
DOT  216     84 KTR = 2
DOT  217        CALL COLSOL (A(N10),A(N11),A(N9),NUMNP,MB,NWK,KTR)
DOT  218  C
DOT  219  C     Q-VECTOR IS NOW T-VECTOR. SET T(I)=Q(I) AND Q(I)=0.
DOT  220  C
DOT  221        DO 85 I=1,NUMNP
DOT  222        IT = N8  + I - 1
DOT  223        IQ = N11 + I - 1
DOT  224        A(IT) = A(IQ)
DOT  225     85 A(IQ) = 0.0
DOT  226  C
DOT  227  C     PRINT AND/OR PUNCH THE NODAL TEMPERATURE DISTRIBUTION,
DOT  228  C     IF REQUESTED, AT THIS TIME STEP
DOT  229  C
DOT  230        K = MOD(KSTEP,NPRINT)
DOT  231        IF(K.NE.0) GO TO 90
DOT  232        CALL OUT (A(N8),NUMNP,TIME,KSTEP)
DOT  233     90 IF(KP.EQ.0) GO TO 92
DOT  234        L = MOD(KSTEP,KP)
DOT  235        IF(L.NE.0) GO TO 92
DOT  236        CALL PTEMP (A(N8),TIME,NUMNP)
DOT  237  C
DOT  238  C     COMPUTE THE NEW THERMAL RESISTIVITY VECTOR (E)
DOT  239  C
DOT  240     92 IF(KST.EQ.-1) GO TO 95
DOT  241  C
DOT  242        CALL FORME (A(N8),A(N13),A(N12),NUMNP)
DOT  243  C
DOT  244  C     CHECK FOR FINAL TIME STEP
DOT  245  C
DOT  246     95 IF(KSTEP.LT.NDT) GO TO 100
DOT  247  C
DOT  248        GO TO 200
DOT  249  C
DOT  250        END
```

```
DOTI   1        SUBROUTINE DOTI
DOTI   2  C
DOTI   3        COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
DOTI   4        COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
DOTI   5        COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
DOTI   6        COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST
DOTI   7        COMMON /JUNK  / HED(18),MTOT,NLINE
DOTI   8        COMMON /NBC   / NNBC,NBCF,NPTM
DOTI   9        COMMON  A(1)
DOTI  10        DIMENSION  MOD(2)
DOTI  11        DATA  MOD/5HSTART,5HSTOP /
DOTI  12  C
DOTI  13  C     ========================
DOTI  14  C     READ CONTROL INFORMATION
DOTI  15  C     ========================
DOTI  16  C
DOTI  17     10 READ (5,1000) MODE,HED
DOTI  18        IF(MODE.EQ.MOD(2)) STOP
DOTI  19        IF(MODE.EQ.MOD(1)) GO TO 20
DOTI  20        WRITE(6,3000)
DOTI  21        GO TO 10
DOTI  22  C
DOTI  23     20 READ (5,1001) NUMNP,NEG,NUMEST,MODEX
```

```
C
      IF(NUMEST.EQ.0) NUMEST = 4000
      IF(NUMNP.GT.0) GO TO 30
      WRITE(6,3001)
      STOP
   30 IF(NEG.GT.0) GO TO 40
      WRITE(6,3002)
      STOP
C
   40 CALL TITLE (HED)
      WRITE(6,2000) NUMNP,NEG,NUMEST,MODEX
      N_INE = 17
C
C
C     ****************************************************************
C
C     B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C          ARRAY  -----------DESCRIPTION-----------   DIMENSION
C     N1   X      NODAL X-COORDINATES                 NUMNP
C     N2   Y      NODAL Y-COORDINATES                 NUMNP
C     N3   TFN    TIME VALUES AT POINTS               NPTM*NBCF
C     N4   FN     FUNCTION VALUES AT POINTS           NPTM*NBCF
C     N5   NPTS   NUMBER OF FUNCTION INPUT POINTS     NBCF
C     N6   NOD    NODES HAVING PRESCRIBED BC          NNBC
C     N7   KODE   BOUNDARY CONDITION TYPES            NNBC
C     N8   NFN    BC FUNCTION NUMBERS                 NNBC
C     N9   TQ     BOUNDARY VALUE AMPLITUDES           NNBC
C     N10  T      NODAL TEMPERATURES                  NUMNP
C     N11  MAXA   ADDRESSES OF (XK) DIAGONAL ELTS.    NUMNP+1
C     N12  MHT    ACTIVE COLUMN HEIGHTS               NUMNP
C
C     ****************************************************************
C
C     ================================
C     READ NODAL POINT COORDINATE DATA
C     ================================
C
      N1  =  1   +  NUMEST
      N2  =  N1  +  NUMNP
      N3  =  N2  +  NUMNP
      IF(N3.GT.MTOT) CALL ERROR (N3-MTOT)
C
      CALL COORD (A(N1),A(N2),NUMNP)
C
C     ===================================================
C     READ SOLUTION TIME AND TEMPERATURE CONTROL INFORMATION
C     ===================================================
C
      READ (5,1002) KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,KP
C
      IF(NDT   .EQ.0) NDT    = 1
      IF(NPRINT.EQ.0) NPRINT = 1
      CALL TITLE (HED)
      WRITE(6,2001) KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,KP
      NLINE = 22
C
C     ===================================
C     CALCULATE AND STORE BC TIME FUNCTIONS
C     ===================================
C
      READ (5,1003) NBCF,NPTM,NNBC
      WRITE(5,2002) NBCF,NPTM
      NLINE = NLINE + 9
      IF(NBCF.GT.0) GO TO 50
      N6  = N3
      GO TO 60
   50 N4  =  N3  +  NPTM*NBCF
      N5  =  N4  +  NPTM*NBCF
      N6  =  N5  +  NBCF
      IF(N6.GT.MTOT) CALL ERROR (N6-MTOT)
C
      CALL FUNC (A(N3),A(N4),A(N5),NPTM)
C
C     ==================================
C     READ NODAL POINT BOUNDARY CONDITIONS
C     ==================================
C
   60 CALL TITLE (HED)
      WRITE(6,2003) NNBC
      NLINE = 10
      IF(NNBC.GT.0) GO TO 70
      N10 = N6
      GO TO 80
   70 N7  =  N6  +  NNBC
      N8  =  N7  +  NNBC
      N9  =  N8  +  NNBC
      N10 =  N9  +  NNBC
      IF(N10.GT.MTOT) CALL ERROR (N10-MTOT)
C
      CALL NPBC (A(N6),A(N7),A(N8),A(N9),NUMNP)
C
C     ====================================
C     DEFINE INITIAL TEMPERATURE DISTRIBUTION
C     ====================================
C
   80 N11 =  N10 +  NUMNP
      IF(N11.GT.MTOT) CALL ERROR (N11-MTOT)
C
      CALL INITAL (A(N10),TAMB,NUMNP)
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(A5,3X,18A4)
 1001 FORMAT(4I5)
 1C02 FORMAT(2I5,3F10.0,3I5)
 1003 FORMAT(3I5)
 2000 FORMAT(20(1H*)/20H CONTROL INFORMATION/20(1H*)///
     1       34H     NUMBER OF NODAL POINTS ..... = I5/
     2       34H     NUMBER OF ELEMENT GROUPS ... = I5/
     3       34H     MAX. ELEMENT GROUP STORAGE . = I5//
     4       34H     SOLUTION MODE .............. = I5/
     4       23H       EQ. 0, DATA CHECK/6X,16HEQ. 1, EXECUTION///)
 2001 FORMAT(50(1H*)/38H SOLUTION TIME AND TEMPERATURE CONTROL,
     1       12H INFORMATION/50(1H*)///
     2       48H     CODE FOR STEADY STATE, TRANSIENT ANALYSIS  = I5/
     3       35H       EQ. -1, STEADY STATE ANALYSIS/
     4       32H       EQ.  0, TRANSIENT ANALYSIS//
     5       48H     NUMBER OF SOLUTION TIME STEPS ............ = I5/
     6       48H     SOLUTION TIME STEP INCREMENT ............. = F10.4//
     7       48H     TIME AT SOLUTION START ................... = F10.4/
     8       48H     AMBIENT TEMPERATURE ...................... = F10.4/
     9       48H     OUTPUT PRINT INTERVAL .................... = I5/
     A       48H     EQUATION REFORMATION INTERVAL ............ = I5/
     B       48H     OUTPUT PUNCH INTERVAL .................... = I5///)
 2002 FORMAT(25(1H*)/25H TIME DEPENDENT FUNCTIONS/25(1H*)///
     1       43H     NUMBER OF TIME DEPENDENT FUNCTIONS ....... = I5/
     2       48H     MAXIMUM NUMBER OF (F(T),T) POINTS ........ = I5//)
 2003 FORMAT(32(1H*)/32H NODAL POINT BOUNDARY CONDITIONS/32(1H*)///
     1       34H     NUMBER OF NODES WITH SPECIFIED/
     2       38H     TEMPERATURES AND/OR FLUXES ..... = I5/)
 3OCC FORMAT(//51H **ERROR**  PROPLEM DECK MUST BEGIN WITH START CARD)
 3001 FORMAT(//49H **ERROR**  NO. OF NODAL POINTS MUST BE .GT. ZERO)
 3002 FORMAT(//51H **ERROR**  NO. OF ELEMENT GROUPS MUST BE .GT. ZERO)
C
      RETURN
      END
      SUBFOUTINE COORD (X,Y,NUMNP)
C
C     ******************************************************************
C     THIS ROUTINE READS AND GENERATES THE GLOBAL NODAL POINT
C     COORDINATE DATA FOR 4- TO 8-NODE ISOPARAMETRIC ELEMENTS
C     ******************************************************************
C
      DIMENSION  X(1),Y(1)
      COMMON /JUNK  / HED(18),MTOT,NLINE
C
C     READ OR GENERATE NODAL POINT DATA
C
      WRITE(6,2000)
      WRITE(6,2001)
      NLINE = NLINE + 12
      NOLD  = 0
C
   10 READ (5,1000) N,X(N),Y(N),KN,JPR
      IF(N.EQ.1) IPR=JPR
      IF(NLINE.LT.55) GO TO 15
      CALL TITLE (HED)
      WRITE(6,2001)
      NLINE = 10
C
   15 WRITE(6,2002) N,X(N),Y(N),KN
      NLINE = NLINE + 1
      IF(NOLD.EQ.0) GO TO 30
C
C     CHECK IF GENERATION IS REQUIRED
C
      IF(KNOLD.EQ.0) GO TO 30
      NUM  = (N-NOLD)/KNOLD
      NUMN = NUM - 1
      RNUM = NUM
      DX   = (X(N)-X(NOLD))/RNUM
      DY   = (Y(N)-Y(NOLD))/RNUM
      K    = NOLD
      DO 20 J=1,NUMN
      KK   = K
      K    = K + KNOLD
      X(K) = X(KK) + DX
   20 Y(K) = Y(KK) + DY
C
   30 NOLD = N
      KNOLD= KN
      IF(N.NE.NUMNP) GO TO 10
C
      IF(IPR.EQ.1) GO TO 200
C
C     PRINT ALL NODAL POINT DATA
C
      CALL TITLE (HED)
      WRITE(6,2003)
      NLINE = 9
      NROW = NUMNP/3 + 1
      NR = 0
C
      DO 100 I=1,NUMNP,3
      NR = NR + 1
      IP = I + 2
      IF(NR.EQ.NROW) IP = NUMNP
      IF(NLINE.LT.55) GO TO 50
      CALL TITLE (HED)
      WRITE(6,2003)
      NLINE = 9
   50 WRITE(6,2004) (N,X(N),Y(N),N=I,IP)
  100 NLINE = NLINE + 1
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(I5,5X,2F10.0,I5,I1)
 2000 FORMAT(28(1H*)/28H NODAL POINT COORDINATE DATA/28(1H*)/)
 2001 FORMAT(19(1H*)/19H A. INPUT NODE DATA/19(1H*)///
     1        4X,4HNODE,5X,7HX-COORD,5X,7HY-COORD,5X,4HDIFF/)
 2002 FORMAT(3X,I5,2F12.3,3X,I5)
 2003 FORMAT(23(1H*)/23H B. GENERATED NODE DATA/23(1H*)///
     1        3(4X,4HNODE,5X,7HX-COORD,5X,7HY-COORD,5X)/)
 2004 FORMAT(3(3X,I5,2F12.3,5X))
C
  200 RETURN
      END
```

```
      SUBROUTINE FUNC (TFN,FN,NPTS,NPTM1)
C
C     ******************************************************************
C     DEFINE ALL BOUNDARY CONDITION FUNCTIONS
C     ******************************************************************
C
      DIMENSION  TFN(NPTM1,1),FN(NPTM1,1),NPTS(1)
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      COMMON /WORK  / FORM(4),WORK(196)
C
      WRITE(6,2001)
      NLINE = NLINE + 3
C
      DO 100 L=1,NBCF
      READ (5,1000) NC,NPTS(NC)
      WRITE(6,2002) NC,NPTS(NC)
      NLINE = NLINE + 1
      IF(NPTS(NC).GE.2.AND.NPTS(NC).LE.NPTM) GO TO 20
      WRITE(6,3000)
      STOP
C
C     READ TIME FUNCTION VERSUS TIME TABLE
C
   20 NT   = NPTS(NC)
      READ (5,1001) (TFN(K,NC),FN(K,NC),K=1,NT)
C
C     CHECK THAT TIME POINTS ARE IN INCREASING ORDER
C
      TOLD = -1.
      DO 30 K=1,NT
      IF(TFN(K,NC).GT.TOLD) GO TO 30
      WRITE(6,3001)
      STOP
   30 TOLD = TFN(K,NC)
C
      IF(TFN(1,NC).LE.TSTART) GO TO 40
      WRITE(6,3002)
      STOP
   40 TEND = TSTART + NDT*DT
      IF(TFN(NT,NC).GE.TEND) GO TO 50
      WRITE(6,3003)
      STOP
C
   50 DO 70 K=1,NT
      IF(NLINE.LT.55) GO TO 60
      CALL TITLE (HED)
      WRITE(6,2000)
      WRITE(6,2001)
      NLINE = 10
   60 WRITE(6,2003) K,TFN(K,NC),FN(K,NC)
   70 NLINE = NLINE + 1
C
  100 CONTINUE
C
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(2I5)
 1001 FORMAT(8F10.0)
 2000 FORMAT(25(1H*)/25H TIME DEPENDENT FUNCTIONS/25(1H*)//)
 2001 FORMAT(4X,8HFUNCTION,4X,9HNUMBER OF,6X,10HTIME POINT,4X,4HTIME,
     1       4X,8HFUNCTION/5X,6HNUMBER,4X,11HTIME POINTS,7X,6HNUMBER,
     2       5X,5HVALUE,5X,5HVALUE/)
 2002 FORMAT(4X,I5,7X,I5)
 2003 FORMAT(31X,I6,2F12.3)
 3000 FORMAT(//49H **ERROR**  (NPTS) MUST BE .GE. 2 AND .LE. (NPTM))
 3001 FORMAT(//52H **ERROR**  BC FUNCTION TIME POINTS ARE OUT OF ORDER)
 3002 FORMAT(//40H **ERROR**  TFN(N,1) MUST BE .EQ. TSTART)
 3003 FORMAT(//43H **ERROR**  TFN(N,NPTS) MUST BE .GE. NDT*DT)
C
      RETURN
      END
```

```
      SUBROUTINE NPBC (NOD,KODE,NFN,TQ,NUMNP)
C
C     ******************************************************************
C     INPUT NODAL POINT BOUNDARY CONDITIONS
C     ******************************************************************
C
      DIMENSION  NOD(1),KODE(1),NFN(1),TQ(1)
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      DIMENSION  AST(2)
      DATA  AST/2H  ,2H */
C
      WRITE(6,2001)
      WRITE(6,2002)
      NLINE = NLINE + 7
      N = 1
C
  100 READ (5,1000) M,ND,KD,NF,TQF,KG
C
      IF(KG.EQ.0) KG=1
      IF(ND.GE.1.AND.ND.LE.NUMNP) GO TO 105
      WRITE(6,3000)
      STOP
  105 IF(NF.LE.NBCF) GO TO 110
      WRITE(6,3001)
      STOP
C
  110 IF(M-N) 280,120,200
C
C     SAVE INFORMATION FOR GENERATION OF ADDITIONAL NODES
C
  120 NDM  = ND
      KDM  = KD
      NFM  = NF
      TQM  = TQF
      KKK  = KG
      ASTT = AST(1)
C
C     STORE PERMANENT INFORMATION
C
  200 NOD(N)   = NDM
      KODE(N) = KDM
      NFN(N)  = NFM
      TQ(N)   = TQM
C
      IF(NLINE.LT.35) GO TO 250
      CALL TITLE (HED)
      WRITE(6,2000)
      WRITE(6,2002)
      NLINE = NLINE + 10
C
  250 WRITE(6,2003) ASTT,N,NDM,KDM,NFM,TQM
      NLINE = NLINE + 1
      IF(N.EQ.NNBC) GO TO 300
C
      N = N + 1
C
C     CHECK IF NODAL DATA IS TO BE STORED FOR CURRENT NODE
C
      IF(N.EQ.M) GO TO 120
C
C     GENERATE NEXT NODE NUMBER
C
      NDM = NDM + KKK
C
C     CHECK IF NEXT NODE CARD IS TO BE READ
C
      ASTT = AST(2)
      IF(N.GT.M) GO TO 100
C
C     GENERATE INFORMATION FOR NEXT NODE
C
      GO TO 200
C
  280 WRITE(6,3002) M
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(4I5,F10.0,I5)
 2000 FORMAT(32(1H*)/32H NODAL POINT BOUNDARY CONDITIONS/32(1H*)//)
 2001 FORMAT(3X,50H BOUNDARY CONDITION TYPE = 0 : EXTERNALLY SUPPLIED,
     1       12H HEAT FLUX Q/28X,
     2       36H= 1 : PRESCRIBED NODAL TEMPERATURE T//)
 2002 FORMAT(4X,6HBC NO.,5X,4HNODE,5X,7HBC TYPE,5X,8HFUNCTION,6X,
     1       6HT OR Q/16X,3HNO.,18X,6HNUMBER,7X,6HFACTOR/)
 2003 FORMAT(2X,A2,I4,6X,I5,8X,I1,8X,I5,3X,F12.3)
 3000 FORMAT(//49H **ERROR**  INVALID BOUNDARY CONDITION NODE NO. = I5)
 3001 FORMAT(//38H **ERROR**  NFN(N) MUST BE .LE. (NBCF))
 3002 FORMAT(//26H **ERROR**  ELEMENT CARD =I5,16H OUT OF SEQUENCE)
C
  300 RETURN
      END
```

```
      SUBROUTINE INITAL (T,TAMB,NUMNP)
C
C     ******************************************************************
C     INITIALIZE NODAL POINT TEMPERATURE DISTRIBUTION
C     ******************************************************************
C
      DIMENSION  T(NUMNP)
C
      READ (5,1000) ICON
      WRITE(6,2000) ICON
C
```

```
INIT  12           DO 100 I=1,NUMNP
INIT  13       100 T(I) = TAMB
INIT  14     C
INIT  15           IF(ICON.NE.0) GO TO 200
INIT  16           WRITE(6,2001) TAMB
INIT  17           GO TO 300
INIT  18     C
INIT  19       200 WRITE(6,2002)
INIT  20           READ (5,1001) (ND,T(ND),J=1,ICON)
INIT  21           WRITE(6,2003) (N,T(N),N=1,NUMNP)
INIT  22     C
INIT  23     C     FORMAT STATEMENTS
INIT  24     C
INIT  25      1000 FORMAT(I5)
INIT  26      1001 FORMAT(4(I5,5X,F10.0))
INIT  27      2000 FORMAT(///19(1H*)/19H INITIAL CONDITIONS/19(1H*)//
INIT  28         1        51H    INITIAL CONDITION CODE (ICON) ............... = I5//
INIT  29         2        49H       EQ. 0, ALL NODES SET TO AMBIENT TEMPERATURE/
INIT  30         3        51H       EQ. 1, INITIAL CONDITIONS ARE READ FROM CARDS,3/)
INIT  31      2001 FORMAT(4X,47HALL NODES AT UNIFORM INITIAL TEMPERATURE .... =F10.3)
INIT  32      2002 FORMAT(36(1H*)/36H A. INITIAL TEMPERATURE DISTRIBUTION/36(1H*)/)
INIT  33      2003 FORMAT(6(I6,E14.6))
INIT  34     C
INIT  35       300 RETURN
INIT  36           END
```

```
ELCL   1           SUBROUTINE ELCAL
ELCL   2     C
ELCL   3           COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
ELCL   4           COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
ELCL   5           COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST
ELCL   6           COMMON /JUNK  / HED(18),MTOT,NLINE
ELCL   7           COMMON /WORK  / NST(10),WORK(190)
ELCL   8           COMMON  A(1)
ELCL   9           DIMENSION  LABEL(2,2)
ELCL  10           DATA  LABEL/6HAXISYM,6HMETRIC,6HP L A ,6HN A R /
ELCL  11     C
ELCL  12     C     *******************************************************************
ELCL  13     C     THIS ROUTINE CALLS THE APPROPRIATE ELEMENT ROUTINES FOR READING,
ELCL  14     C     GENERATING AND STORING THE ELEMENT DATA
ELCL  15     C
ELCL  16     C     TAPE ALLOCATION:
ELCL  17     C          TAPE 1 - STORES ELEMENT GROUP DATA
ELCL  18     C     *******************************************************************
ELCL  19     C     T W O   D I M E N S I O N A L   F I N I T E   E L E M E N T S
ELCL  20     C
ELCL  21     C        NPAR(1) = 1
ELCL  22     C        NPAR(2) = NUMBER OF TWO DIMENSIONAL ELEMENTS (NEL1)
ELCL  23     C        NPAR(3) = NUMBER OF FIRST ELEMENT IN THIS GROUP (MFST)
ELCL  24     C        NPAR(4) = ELEMENT TYPE CODE (ITYP2D)
ELCL  25     C                  EQ.0, AXISYMMETRIC
ELCL  26     C                  EQ.1, PLANAR
ELCL  27     C        NPAR(5) = MAXIMUM NUMBER OF NODES (MXNODS)
ELCL  28     C        NPAR(6) = NUMERICAL INTEGRATION ORDER (NINT)
ELCL  29     C        NPAR(7) = NUMBER OF DIFFERENT MATERIALS (NUMAT)
ELCL  30     C        NPAR(8) = MAX. NO. TEMPERATURE POINTS IN MATERIAL TABLE (MAXTP)
ELCL  31     C        NPAR(9) = INTERNAL HEAT GENERATION FLAG (IHFLG)
ELCL  32     C     *******************************************************************
ELCL  33     C     C O N V E C T I O N   A N D   R A D I A T I O N   E L E M E N T S
ELCL  34     C
ELCL  35     C        NPAR(1) = 2
ELCL  36     C        NPAR(2) = NUMBER OF CONVECTION BOUNDARY ELEMENTS (NEL2)
ELCL  37     C        NPAR(3) = NUMBER OF FIRST ELEMENT IN THIS GROUP (MFST)
ELCL  38     C        NPAR(4) = ELEMENT TYPE CODE (ITYP)
ELCL  39     C                  EQ.0, AXISYMMETRIC SOLID CONVECTION BC
ELCL  40     C                  EQ.1, PLANAR SOLID CONVECTION BC
ELCL  41     C        NPAR(5) = NUMBER OF DIFFERENT MATERIALS (NUMAT)
ELCL  42     C     *******************************************************************
ELCL  43     C     C O O L I N G   P I P E   E L E M E N T S
ELCL  44     C
ELCL  45     C        NPAR(1) = 3
ELCL  46     C        NPAR(2) = NUMBER OF COOLING PIPE ELEMENTS (NEL3)
ELCL  47     C        NPAR(3) = NUMBER OF FIRST ELEMENT IN THIS GROUP (MFST)
ELCL  48     C     *******************************************************************
ELCL  49     C
ELCL  50     C     ZERO ACTIVE COLUMN HEIGHT ARRAY (MHT)
ELCL  51     C
ELCL  52           N12 = N11 + NUMNP + 1
ELCL  53           N13 = N12 + NUMNP
ELCL  54           IF(N13.GT.MTOT) CALL ERROR (N13-MTOT)
ELCL  55     C
ELCL  56           DO 5 I=N12,N13
ELCL  57         5 A(I) = 0.0
ELCL  58     C
ELCL  59           REWIND 1
ELCL  60     C
ELCL  61     C     LOOP OVER ALL ELEMENT GROUPS
ELCL  62     C
ELCL  63           DO 100 NG=1,NEG
ELCL  64           CALL TITLE (HED)
ELCL  65           WRITE(6,2000) NG
ELCL  66           NLINE = 7
ELCL  67     C
ELCL  68           READ (5,1000) NPAR
ELCL  69     C
ELCL  70           NGR  = NPAR(1)
ELCL  71     C
ELCL  72           GO TO (1,2,3) NGR
ELCL  73     C
ELCL  74     C     ---------------
ELCL  75     C     ELEMENT GROUP 1
ELCL  76     C     ---------------
ELCL  77     C
ELCL  78         1 IF(NPAR(2).GT.0) GO TO 10
ELCL  79           WRITE(6,3000)
ELCL  80           STOP
ELCL  81        10 IF(NPAR(6).LE.4) GO TO 20
ELCL  82           WRITE(6,3001)
ELCL  83           STOP
ELCL  84        20 IF(NPAR(3).EQ.0) NPAR(3) = 1
ELCL  85           IF(NPAR(5).EQ.0) NPAR(5) = 4
ELCL  86           IF(NPAR(6).EQ.0) NPAR(6) = 2
ELCL  87           IF(NPAR(7).EQ.0) NPAR(7) = 1
ELCL  88           IF(NPAR(8).EQ.0) NPAR(8) = 1
ELCL  89           IT = NPAR(4) + 1
ELCL  90     C
ELCL  91           WRITE(6,2001) NGR,(LABEL(I,IT),I=1,2),NPAR(2),NPAR(3),NPAR(5),
ELCL  92          1               NPAR(6),NPAR(7),NPAR(8),NPAR(9)
ELCL  93           CALL ADRS1
ELCL  94           GO TO 50
ELCL  95     C
ELCL  96     C     ---------------
ELCL  97     C     ELEMENT GROUP 2
ELCL  98     C     ---------------
ELCL  99     C
ELCL 100         2 IF(NPAR(2).GT.0) GO TO 30
ELCL 101           WRITE(6,3000)
ELCL 102           STOP
ELCL 103        30 IF(NPAR(3).EQ.0) NPAR(3) = 1
ELCL 104           IF(NPAR(5).EQ.0) NPAR(5) = 1
ELCL 105           IT = NPAR(4) + 1
ELCL 106     C
ELCL 107           WRITE(6,2002) NGR,(LABEL(I,IT),I=1,2),NPAR(2),NPAR(3),NPAR(5)
ELCL 108           CALL ADRS2
```

```
      GO TO 50
C
C     ---------------
C     ELEMENT GROUP 3
C     ---------------
C
    3 IF(NPAR(2).GT.0) GO TO 40
      WRITE(6,3000)
      STOP
   40 IF(NPAR(3).EQ.0) NPAR(3) = 1
C
      WRITE(6,2003) NGR,NPAR(2),NPAR(3)
      NLINE = NLINE + 6
      CALL ADRS3
C
   50 IF(MIDEST.GT.MAXEST) MAXEST = MIDEST
      IF(MIDEST.LE.NUMEST) GO TO 60
      GO TO 100
C
C     STORE ALL ELEMENT GROUP INFORMATION ONTO TAPE 1
C
   60 WRITE(1) MIDEST,NPAR,NST,(A(I),I=1,MIDEST)
C
  100 CONTINUE
C
      IF(MAXEST.LE.NUMEST) GO TO 300
      WRITE(6,3002) MAXEST
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(10I5)
 2000 FORMAT(23(1H*)/20H ELEMENT DATA, GROUP,I3/23(1H*)//)
 2001 FORMAT(26H ELEMENT GROUP INDICATOR =I3,18H (TWO DIMENSIONAL ,2A6,
     1       10H ELEMENTS)///
     2       38H    NUMBER OF ELEMENTS ............. = I5/
     3       38H    NO. OF FIRST ELEMENT IN GROUP .. = I5/
     4       38H    MAX. NO. OF NODES PER ELEMENT .. = I5/
     5       38H    NUMERICAL INTEGRATION ORDER .... = I5/
     6       38H    NUMBER OF DIFFERENT MATERIALS .. = I5/
     7       38H    MAX. NO. OF TEMPERATURE POINTS . = I5/
     8       38H    INTERNAL HEAT GENERATION FLAG .. = I5/
     9       31H      EQ. 0, NO HEAT GENERATION/
     A       35H      EQ. 1, HEAT GENERATION EXISTS/)
 2002 FORMAT(26H ELEMENT GROUP INDICATOR =I3,
     1       2H (,2A6,36H SOLID CONVECTION BOUNDARY ELEMENTS)///
     2       38H    NUMBER OF ELEMENTS ............. = I5/
     3       38H    NO. OF FIRST ELEMENT IN GROUP .. = I5/
     4       38H    NUMBER OF DIFFERENT MATERIALS .. = I5/)
 2003 FORMAT(26H ELEMENT GROUP INDICATOR =I3,
     1       24H (COOLING PIPE ELEMENTS)///
     2       38H    NUMBER OF ELEMENTS ............. = I5/
     3       38H    NO. OF FIRST ELEMENT IN GROUP .. = I5/)
 3000 FORMAT(//51H **ERROR**  NO. OF ELTS. IN GROUP MUST BE .GT. ZERO)
 3001 FORMAT(//53H **ERROR**  NO. OF INTEGRATION PTS. MUST BE .LE. FOUR)
 3002 FORMAT(//41H **ERROR**  (NUMEST) MUST BE INCREASED TO I6)
C
  300 RETURN
      END
```

```
      SUBROUTINE ADRS1
C
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,K3C
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST
      COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
C
C     *****************************************************************
C
C     B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C            ARRAY  -----------DESCRIPTION-----------    DIMENSION
C     M1     LM     ELEMENT CONNECTIVITY ARRAY           MXNODS*NEL1
C     M2     XY     ELEMENT COORDINATES                  2*MXNODS*NEL1
C     M3     IELT   NO. OF NODES DESCRIBING ELEMENT      NEL1
C     M4     NOD5   MIDSIDE NODES LOCATION ARRAY         ND5DIM*NEL1
C     M5     MATP   MATERIAL PROPERTY SET NUMBERS        NEL1
C     M6     NTC    NO. OF MATERIAL TEMPERATURE CARDS    NUMAT
C     M7     TPROP  MATERIAL PROPERTIES TABLE:           MAXTP*5*NUMAT
C                   T - K(11),K(22),K(12),C
C     M8     DENS   DENSITY                              NUMAT
C     M9     NFHG   FUNCTION ID FOR HEAT GENERATION      NUMAT
C     M10    HGSC   HEAT GENERATION SCALE FACTOR         NUMAT
C
C     *****************************************************************
C
      NEL1   = NPAR(2)
      MXNODS = NPAR(5)
      NUMAT  = NPAR(7)
      MAXTP  = NPAR(8)
      NDM    = 2*MXNODS
      ND5DIM = MXNODS-4
C
      M1     = 1
      M2     = M1  + MXNODS*NEL1
      M3     = M2  + NDM*NEL1
      M4     = M3  + NEL1
      M5     = M4  + ND5DIM*NEL1
      M6     = M5  + NEL1
      M7     = M6  + NUMAT
      M8     = M7  + MAXTP*5*NUMAT
      M9     = M8  + NUMAT
      M10    = M9  + NUMAT
      M11    = M10 + NUMAT
      NLAST  = M11 - 1
C
      WRITE(6,2000) NLAST
      MIDEST = NLAST
      IF(ND5DIM.EQ.0) ND5DIM = 1
C
      CALL ELGR1 (A(N1),A(N2),A(N12),A(M1),A(M2),A(M3),A(M4),A(M5),
     1            A(M6),A(M7),A(M8),A(M9),A(M10),MXNODS,NDM,ND5DIM,
     2            MAXTP,NUMAT)
C
 2000 FORMAT(38H    LENGTH OF ELEMENT INFORMATION .. = I5///)
C
      RETURN
      END
      SUBROUTINE ELGR1 (X,Y,MHT,LM,XY,IELT,NOD5,MATP,NTC,TPROP,DENS,
     1                  NFHG,HGSC,MXNODS,NDM,ND5DIM,MAXTP,NUMAT)
C
C     ******************************************************************
C     INPUT INFORMATION FOR 4- TO 8-NODE ISOPARAMETRIC ELEMENTS
C     ******************************************************************
C
      DIMENSION  X(1),Y(1),MHT(1),LM(MXNODS,1),XY(NDM,1),IELT(1),
     1           NOD5(ND5DIM,1),MATP(1),NTC(1),TPROP(MAXTP,5,NUMAT),
     2           DENS(1),NFHG(1),HGSC(1)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /WORK  / DUM(10),NOD(8),NODM(8),NOD5M(8),WORK(166)
      DIMENSION  AST(2)
      DATA  AST/2H  ,2H */
C
      NEL1  = NPAR(2)
      MFST  = NPAR(3)
C
C     ======================================
C     READ AND PRINT MATERIAL PROPERTY TABLE
C     ======================================
C
      CALL TITLE (HED)
      WRITE(6,2000) NG
C
      DO 50 N=1,NUMAT
      READ (5,1000) M,DENS(M),NTC(M),NFHG(M),HGSC(M)
      NTP = NTC(M)
      IF(NTP.EQ.0) NTC(M) = 1
      WRITE(6,2001) M,DENS(M),NTC(M),NFHG(M),HGSC(M)
      IF(NTP.LE.MAXTP) GO TO 10
      WRITE(6,3000)
      STOP
C
C     READ TEMPERATURE DEPENDENT PROPERTIES
C
   10 TOLD = -1.0E+6
      DO 20 K=1,NTP
      READ (5,1001)   (TPROP(K,J,M),J=1,5)
      WRITE(6,2002) K,(TPROP(K,J,M),J=1,5)
C
C     TEMPERATURES MUST BE IN ASCENDING ORDER
C
      IF(TPROP(K,1,M).GT.TOLD) GO TO 20
      WRITE(6,3001)
      STOP
   20 TOLD = TPROP(K,1,M)
   50 CONTINUE
C
C     ====================================
C     READ AND GENERATE ELEMENT INFORMATION
C     ====================================
C
      CALL TITLE (HED)
      WRITE(6,2003) NG
      WRITE(6,2004) (I,I=1,8)
      NLINE = 10
      N     = 1
      IMEM  = MFST
      NLAST = MFST + NEL1 - 1
C
  100 READ (5,1002) M,NOD,MTYP,IEL,KG
C
      IF(MTYP.EQ.0) MTYP = 1
      IF(IEL .EQ.0) IEL  = MXNODS
      IF(KG  .EQ.0) KG   = 1
      IF(MXNODS.GE.IEL) GO TO 110
      WRITE(6,3002) M
      STOP
C
  110 IF(M-IMEM) 280,120,200
C
C     SAVE ELEMENT INFORMATION FOR GENERATION OF ADDITIONAL ELEMENTS
C
  120 DO 130 I=1,8
  130 NODM(I)=NOD(I)
      IF(IEL.EQ.4) GO TO 150
      II=0
      DO 140 I=5,8
      NN=NOD(I)
      IF(NN.EQ.0) GO TO 140
      II=II + 1
      NOD5M(II)=I
  140 CONTINUE
C
  150 MTYPE = MTYP
      IELM  = IEL
      KKK   = KG
      ASTT  = AST(1)
C
C     STORE PERMANENT ELEMENT INFORMATION
C
  200 I2 = 0
      DO 230 I=1,IELM
      IF(I.LE.4) GO TO 210
      JJ = NOD5M(I-4)
      II = NODM(JJ)
      GO TO 220
  210 II = NODM(I)
  220 LM(I,N) = II
      I2 = I2 + 2
      XY(I2-1,N) = X(II)
  230 XY(I2  ,N) = Y(II)
C
      IELT(N) = IELM
      MATP(N) = MTYPE
      IF(IELM.EQ.4) GO TO 250
      NN=IELM - 4
      DO 240 I=1,NN
  240 NOD5(I,N)=NOD5M(I)
C
C     UPDATE COLUMN HEIGHTS AND BANDWIDTH
C
  250 CALL COLHT (MHT,IELM,LM(1,N))
C
      IF(NLINE.LT.55)  GO TO 260
      CALL TITLE (HED)
      WRITE(6,2003) NG
      WRITE(6,2004) (I,I=1,8)
      NLINE = 10
C
  260 WRITE(6,2005) ASTT,IMEM,NODM,MTYPE,IELM
      NLINE = NLINE + 1
      IF(IMEM.EQ.NLAST)  GO TO 300
C
      N = N + 1
      IMEM = IMEM + 1
C
C     CHECK IF ELEMENT DATA IS TO BE STORED FOR CURRENT ELEMENT
C
      IF(IMEM.EQ.M) GO TO 120
C
C     GENERATE NODE NUMBERS FOR NEXT ELEMENT
C
      DO 270 I=1,8
      IF(NODM(I).EQ.0) GO TO 270
      NODM(I)=NODM(I)+KKK
  270 CONTINUE
C
```

```
C      CHECK IF NEXT ELEMENT CARD IS TO BE READ
C
       ASTT=AST(2)
       IF(IMEM.GT.M) GO TO 100
C
C      GENERATE INFORMATION FOR NEXT ELEMENT
C
       GO TO 200
C
  280 WRITE(6,3003) M
       STOP
C
C      FORMAT STATEMENTS
C
 1000 FORMAT(I5,5X,F10.0,2I5,F10.0)
 1001 FORMAT(5F10.0)
 1002 FORMAT(12I5)
 2000 FORMAT(36(1H*)/33H MATERIAL PROPERTIES TABLE, GROUP,I3/36(1H*))
 2001 FORMAT(//31H    MATERIAL NUMBER .......... = I5/
     1          31H    DENSITY ................. = E14.4/
     2          31H    NO. OF TEMPERATURE POINTS = I5/
     3          31H    HEAT GENERATION FUNCTION  = I5/
     4          31H    HEAT GENERATION SCALE ... = E14.4///
     5        4X,5HPOINT,4X,11HTEMPERATURE,6X,9(1H-),7HTHERMAL,
     6        13H CONDUCTIVITY,9(1H-),4X,13HSPECIFIC HEAT/4X,6HNUMBER,
     7        23X,5HK(11),9X,5HK(22),9X,5HK(12),8X,8HCAPACITY/)
 2002 FORMAT(3X,I5,2X,E14.3,1X,3E14.3,E15.3)
 2003 FORMAT(30(1H*)/27H ELEMENT INFORMATION, GROUP,I3/30(1H*)//)
 2004 FORMAT(4X,4HELT.,3X,10(1H-),12HNODE NUMBERS,10(1H-),3X,5HMATL.,3X,
     1        6HNO. OF/5X,3HNO.,3X,8(3X,I1),4X,3HNO.,5X,5HNODES/)
 2005 FORMAT(A2,I5,4X,8I4,1X,I5,4X,I5)
 3000 FORMAT(//53H **ERROR**  NO. OF TEMPERATURE POINTS EXCEEDS MAXIMUM)
 3001 FORMAT(//51H **ERROR**  TEMPERATURES MUST BE IN ASCENDING ORDER)
 3002 FORMAT(//10H **ELEMENT,I5,34H EXCEEDS MAXIMUM NUMBER OF NODES**)
 3003 FORMAT(//26H **ERROR**  ELEMENT CARD =I5,16H OUT OF SEQUENCE)
C
  300 RETURN
       END
```

```
       SUBROUTINE ADRS2
C
       COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
       COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
       COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST
       COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
       COMMON  A(1)
C
C      ******************************************************************
C
C      B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C           ARRAY  -----------DESCRIPTION-----------   DIMENSION
C      M1   LM     ELEMENT CONNECTIVITY ARRAY          2*NEL2
C      M2   XX     ELEMENT X-COORDINATES               2*NEL2
C      M3   CL     ELEMENT LENGTHS                     NEL2
C      M4   MATP   MATERIAL PROPERTY SET NUMBERS       NEL2
C      M5   SB     STEFAN-BOLTZMANN CONSTANT           1
C      M6   TSHIFT SHIFT FOR ABSOLUTE TEMPERATURE      1
C      M7   NFH    FUNCTION ID FOR H(T) VARIATION      NUMAT
C      M8   NFTE   FUNCTION ID FOR (TE) VARIATION      NUMAT
C      M9   PROP   MATERIAL PROPERTIES TABLE           5*NUMAT
C
C      ******************************************************************
C
       NEL2  = NPAR(2)
       NUMAT = NPAR(5)
C
       M1    = 1
       M2    = M1  + 2*NEL2
       M3    = M2  + 2*NEL2
       M4    = M3  + NEL2
       M5    = M4  + NEL2
       M6    = M5  + 1
       M7    = M6  + 1
       M8    = M7  + NUMAT
       M9    = M8  + NUMAT
       M10   = M9  + 5*NUMAT
       NLAST= M10 - 1
C
       WRITE(6,2000) NLAST
       MIDEST = NLAST
C
       CALL ELGR2 (A(N1),A(N2),A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),
     1               A(M7),A(M8),A(M9),NUMAT)
C
 2000 FORMAT(38H    LENGTH OF ELEMENT INFORMATION .. = I5///)
C
       RETURN
       END
```

```
       SUBROUTINE ELGR2 (X,Y,LM,XX,CL,MATP,SB,TSHIFT,NFH,NFTE,PROP,NUMAT)
C
C      ******************************************************************
C      INPUT INFORMATION FOR CONVECTION AND RADIATION SURFACE ELEMENTS
C      ******************************************************************
C
       DIMENSION  X(1),Y(1),LM(2,1),XX(2,1),CL(1),MATP(1),NFH(1),
     1            NFTE(1),PROP(5,NUMAT)
C
       COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
       COMMON /JUNK  / HED(18),MTOT,NLINE
       COMMON /NBC   / NNBC,NBCF,NPTM
       COMMON /WORK  / DUM(10),NOD(2),NODM(2),WORK(186)
       DIMENSION  AST(2)
       DATA  AST/2H  ,2H */
C
       NEL2 = NPAR(2)
       MFST = NPAR(3)
C
C      ====================================
C      READ AND PRINT MATERIAL PROPERTY TABLE
C      ====================================
C
       KBC = 0
       WRITE(6,2000) NG
       READ (5,1000) SB,TSHIFT
       WRITE(6,2001) SB,TSHIFT
C
       DO 30 N=1,NUMAT
       READ (5,1001) M,NFH(M),NFTE(M)
       NFS = NFH(M)
       NFT = NFTE(M)
       IF(NFS.NE.0) KBC = 1
       IF(NFS.GE.0.AND.NFS.LE.NBCF) GO TO 10
       WRITE(6,3000)
       STOP
   10 IF(NFT.GE.0.AND.NFT.LE.NBCF) GO TO 20
       WRITE(6,3001)
```

```
      STOP
   20 READ (5,1002) (PROP(J,M),J=1,5)
      V = PROP(3,M)
      IF(V.NE.0.) KBC = 1
   30 WRITE(6,2002) M,NFS,NFT,(PROP(J,M),J=1,5)
C
C     ===================================
C     READ AND GENERATE ELEMENT INFORMATION
C     ===================================
C
      N      = 1
      IMEM   = MFST
      NLAST = MFST + NEL2 - 1
      CALL TITLE (HED)
      WRITE(6,2003) NG
      WRITE(6,2004)
      NLINE = 10
C
  100 READ (5,1003) M,NOD,MTYP,KG
C
      IF(MTYP.EQ.0) MTYP = 1
      IF(KG   .EQ.0) KG    = 1
      II = NOD(1)
      JJ = NOD(2)
C
      IF(M-IMEM) 280,120,200
C
C     SAVE ELEMENT INFORMATION FOR GENERATION OF ADDITIONAL ELEMENTS
C
  120 NODM(1) = II
      NODM(2) = JJ
      MTYPE   = MTYP
      KKK     = KG
      ASTT    = AST(1)
      XL      = SQRT((X(JJ)-X(II))**2 + (Y(JJ)-Y(II))**2)
C
C     STORE PERMANENT ELEMENT INFORMATION
C
  200 DO 230 I=1,2
      IJ = NODM(I)
      LM(I,N) = IJ
  230 XX(I,N) = X(IJ)
C
      CL(N)   = XL
      MATP(N) = MTYPE
C
      IF(NLINE.LT.55) GO TO 250
      CALL TITLE (HED)
      WRITE(6,2003) NG
      WRITE(6,2004)
      NLINE = 10
C
  250 WRITE(6,2005) ASTT,IMEM,NODM,MTYPE
      NLINE = NLINE + 1
      IF(IMEM.EQ.NLAST) GO TO 300
C
      N = N + 1
      IMEM = IMEM + 1
C
C     CHECK IF ELEMENT DATA IS TO BE STORED FOR CURRENT ELEMENT
C
      IF(IMEM.EQ.M) GO TO 120
C
C     GENERATE NODE NUMBERS FOR NEXT ELEMENT
C
      DO 270 I=1,2
  270 NODM(I) = NODM(I) + KKK
C
C     CHECK IF NEXT ELEMENT CARD IS TO BE READ
C
      ASTT = AST(2)
      IF(IMEM.GT.M) GO TO 100
C
C     GENERATE INFORMATION FOR NEXT ELEMENT
C
      GO TO 200
C
  280 WRITE(6,3002) M
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(2E10.0)
 1001 FORMAT(3I5)
 1002 FORMAT(5F10.0)
 1003 FORMAT(5I5)
 2000 FORMAT(36(1H*)/33H MATERIAL PROPERTIES TABLE, GROUP,I3/36(1H*))
 2001 FORMAT(//32H    STEFAN-BOLTZMANN CONSTANT  = E12.4/
     1          32H    ABSOLUTE TEMPERATURE SHIFT = E12.4///
     2        4X,5HMATL.,4X,2(10HCONVECTION,4X,10HT-EXTERNAL,4X),4X,
     3        4HVIEW,8X,8HEXTERNAL,6X,7HSURFACE/5X,3HNO.,6X,
     4        2(8HFUNCTION,6X),2(9HAMPLITUDE,5X),2X,6HFACTOR,6X,
     5        2(10HEMISSIVITY,4X)/)
 2002 FORMAT(2X,I5,7X,I5,9X,I5,7X,5(F10.3,4X))
 2003 FORMAT(30(1H*)/27H ELEMENT INFORMATION, GROUP,I3/30(1H*)//)
 2004 FORMAT(4X,4HELT.,4X,6HI-NODE,4X,6HJ-NODE,4X,5HMATL./5X,
     1        3HNO.,25X,3HNO./)
 2005 FORMAT(A2,I5,5X,I5,5X,I5,3X,I5)
 3000 FORMAT(//38H **ERROR**  NFH(M) MUST BE .LE. (NBCF))
 3001 FORMAT(//39H **ERROR**  NFTE(M) MUST BE .LE. (NBCF))
 3002 FORMAT(//26H **ERROR**  ELEMENT CARD =I5,16H OUT OF SEQUENCE)
C
  300 RETURN
      END
```

```
      SUBROUTINE ADRS3
C
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /WORK  / M1;M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
C
C     ****************************************************************
C
C     B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C           ARRAY  -----------DESCRIPTION-----------   DIMENSION
C     M1    LM     ELEMENT LOCATION ARRAY               NEL3
C     M2    H      EFFECTIVE CONVECTION COEFFICIENT     NEL3
C     M3    NFCP   FUNCTION ID FOR (TW) VARIATION       NEL3
C     M4    TW     COOLING WATER TEMP. AMPLITUDE        NEL3
C     M5    TB     TIME AT START OF COOLING             NEL3
C     M6    TD     TIME AT STOP OF COOLING              NEL3
C
C     ****************************************************************
C
      NEL3 = NPAR(2)
C
      M1    = 1
      M2    = M1   + NEL3
      M3    = M2   + NEL3
      M4    = M3   + NEL3
      M5    = M4   + NEL3
      M6   = M5  + NEL3
      M7   = M6  + NEL3
      NLAST= M7  - 1
C
      WRITE(6,2000) NLAST
      NLINE  = NLINE + 4
      MIDEST = NLAST
C
      CALL ELGR3 (A(M1),A(M2),A(M3),A(M4),A(M5),A(M6))
C
 2000 FORMAT(38H    LENGTH OF ELEMENT INFORMATION .. = I5///)
C
      RETURN
      END
```

```
      SUBROUTINE ELGR3 (LM,H,NFCP,TW,TB,TD)
C
      DIMENSION  LM(1),H(1),NFCP(1),TW(1),TB(1),TD(1)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      DIMENSION  AST(2)
      DATA  AST/2H  ,2H */
C
      NEL3 = NPAR(2)
      MFST = NPAR(3)
C
C     ====================================
C     READ AND GENERATE ELEMENT INFORMATION
C     ====================================
C
      N      = 1
      IMEM   = MFST
      NLAST = MFST + NEL3 - 1
      WRITE(6,2000) NG
      WRITE(6,2001)
      NLINE = NLINE + 8
  100 READ (5,1000) M,NOD,HP,NFC,KG,TWTR,TBTH,TDTH
C
      IF(KG .EQ.0) KG = 1
      IF(NOD.GE.1.AND.NOD.LE.NUMNP) GO TO 10
      WRITE(6,3000) NOD
      STOP
   10 IF(NFC.GE.0.AND.NFC.LE.NBCF) GO TO 110
      WRITE(6,3001)
      STOP
C
  110 IF(M-IMEM) 280,120,200
C
C     SAVE ELEMENT INFORMATION FOR GENERATION OF ADDITIONAL ELEMENTS
C
  120 NODM  = NOD
      HPM   = HP
      NFCM  = NFC
      TWM   = TWTR
      TBORN = TBTH
      TDEAD = TDTH
      KKK   = KG
      ASTT  = AST(1)
C
C     STORE PERMANENT ELEMENT INFORMATION
C
  200 LM(N)    = NODM
      H(N)     = HPM
      NFCP(N) = NFCM
      TW(N)   = TWM
      TB(N)   = TBORN
      TD(N)   = TDEAD
C
      IF(NLINE.LT.55) GO TO 260
      CALL TITLE (HED)
      WRITE(6,2000) NG
      WRITE(6,2001)
      NLINE = 10
C
  260 WRITE(6,2002) ASTT,IMEM,NODM,HPM,NFCM,TWM,TBORN,TDEAD
      NLINE = NLINE + 1
      IF(IMEM.EQ.NLAST) GO TO 300
C
      N = N + 1
      IMEM = IMEM + 1
C.
C     CHECK IF ELEMENT DATA IS TO BE STORED FOR CURRENT ELEMENT
C
      IF(IMEM.EQ.M) GO TO 120
C
C     GENERATE NODE NUMBERS FOR NEXT ELEMENT
C
      NODM = NODM + KKK
C
C     CHECK IF NEXT ELEMENT CARD IS TO BE READ
C
      ASTT = AST(2)
      IF(IMEM.GT.M) GO TO 100
C
C     GENERATE INFORMATION FOR NEXT ELEMENT
C
      GO TO 200
C
  280 WRITE(6,3002) M
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(2I5,F10.0,2I5,3F10.0)
 2000 FORMAT(30(1H*)/27H ELEMENT INFORMATION, GROUP,I3/30(1H*)//)
 2001 FORMAT(4X,4HELT.,4X,4HNODE,5X,10HCONVECTION,5X,4H(TW),5X,4H(TW),
     1       5X,7HTIME OF,5X,7HTIME OF/5X,3HNO.,12X,11HCOEFFICIENT,5X,
     2       4HFUNC,5X,4HAMPL,6X,5HBIRTH,7X,5HDEATH/)
 2002 FORMAT(A2,I5,4X,I5,3X,E12.3,3X,I5,3(F10.2,2X))
 3000 FORMAT(//46H **ERROR**  INVALID COOLING PIPE NODE NUMBER = I5)
 3001 FORMAT(//39H **ERROR**  NFCP(M) MUST BE .LE. (NBCF))
 3002 FORMAT(//26H **ERROR**  ELEMENT CARD =I5,16H OUT OF SEQUENCE)
C
  300 RETURN
      END
```

```
      SUBROUTINE COLHT (MHT,ND,LM)
C
      DIMENSION  LM(1),MHT(1)
C
C     FIND SMALLEST GLOBAL NODE NUMBER (LS) FOR ELEMENT
C
      LS=100000
      DO 100 I=1,ND
      IF (LM(I)) 80,100,80
   80 IF (LM(I)-LS) 90,100,100
   90 LS=LM(I)
```

```
  100 CONTINUE
C
C     COMPUTE COLUMN HEIGHT ABOVE DIAGONAL (ME) AND CHECK IF MAXIMUM
C
      DO 200 I=1,ND
      II=LM(I)
      IF (II.EQ.0) GO TO 200
      ME=II - LS
      IF (ME.GT.MHT(II)) MHT(II)=ME
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE ERROR (N)
C
      WRITE(6,2000) N
 2000 FORMAT(//31H **ERROR**  STORAGE EXCEEDED BY I6)
      STOP
      END
```

```
      SUBROUTINE TITLE (HED)
C
      DIMENSION  HED(18)
C
C     THIS ROUTINE PRINTS THE TITLE CARD AT TOP OF OUTPUT PAGE
C
      WRITE(6,2000) HED
 2000 FORMAT(1H1,18A4,39X,8HOOT 1976/)
      RETURN
      END
```

```
      SUBROUTINE ADRSK (MAXA,MHT,NUMNP,NWK,MA)
C
C     **********************************************************************
C     TO CALCULATE ADDRESSES OF DIAGONAL ELEMENTS IN A
C     BANDED MATRIX WHOSE COLUMN HEIGHTS ARE KNOWN.
C
C     MA   = MAXIMUM BAND WIDTH
C     MHT  = ACTIVE COLUMN HEIGHTS ABOVE DIAGONAL
C     MAXA = ADDRESSES OF DIAGONAL ELEMENTS
C     NWK  = MAXIMUM STORAGE REQUIRED
C     **********************************************************************
C
      DIMENSION  MAXA(1),MHT(1)
C
      MAXA(1) = 1
      MAXA(2) = 2
      MA = 0
      IF (NUMNP.EQ.1) GO TO 100
      DO 10 I=2,NUMNP
      IF (MHT(I).GT.MA) MA = MHT(I)
   10 MAXA(I+1) = MAXA(I) + MHT(I) + 1
  100 MA  = MA + 1
      NWK = MAXA(NUMNP+1) - 1
C
      RETURN
      END
```

```
      SUBROUTINE ASSEMK
C
C     **********************************************************************
C     ASSEMBLE THE EFFECTIVE SYSTEM CONDUCTIVITY MATRIX (K*)
C     **********************************************************************
C
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
      DIMENSION  NST(10)
      EQUIVALENCE  (NST(1),M1)
C
      REWIND 1
      REWIND 2
C
C     LOOP OVER ALL ELEMENT GROUPS
C
      DO 100 NG=1,NEG
C
      READ (1) MIDEST,NPAR,NST,(A(I),I=1,MIDEST)
C
      NGR = NPAR(1)
C
      GO TO (1,2,3) NGR
C
C     ---------------
C     ELEMENT GROUP 1
C     ---------------
C
    1 MXNODS = NPAR(5)
      NUMAT  = NPAR(7)
      MAXTP  = NPAR(8)
      NDM    = 2*MXNODS
      ND5DIM = MXNODS-4
      IF(ND5DIM.EQ.0) ND5DIM = 1
C
      CALL COND1 (A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),A(M7),A(M8),A(N8),
     1            MXNODS,NDM,ND5DIM,MAXTP,NUMAT)
      GO TO 100
C
C     ---------------
C     ELEMENT GROUP 2
C     ---------------
C
    2 NPAR(5) = NUMAT
      CALL COND2 (A(M1),A(M2),A(M3),A(M4),A(M7),A(M9),NUMAT)
      GO TO 100
C
C     ---------------
C     ELEMENT GROUP 3
C     ---------------
C
    3 CALL COND3 (A(M1),A(M2),A(M5),A(M6))
C
  100 CONTINUE
C
```

```
      RETURN
      END
```

```
      SUBROUTINE COND1 (LM,XY,IELT,NOD5,MATP,NTC,TPROP,DENS,T,
     1                  MXNODS,NDM,NOD5DIM,MAXTP,NUMAT)
C
C     ****************************************************************
C     FORM THE EFFECTIVE SYSTEM CONDUCTIVITY MATRIX (K*) AND
C     THE EFFECTIVE HEAT CAPACITY VECTOR (C*)
C     ****************************************************************
C
      DIMENSION  LM(MXNODS,1),XY(NDM,1),IELT(1),NOD5(NOD5DIM,1),MATP(1),
     1           NTC(1),TPROP(MAXTP,5,NUMAT),DENS(1),T(1)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /TODIM / NEL,NODS,MTYPE,NNOD5
      COMMON /WORK  / DUM(10),SK(64),SC(8),HF(8),TNOD(8),WORK(102)
      COMMON  A(1)
C
      NEL1  = NPAR(2)
      IHFLG = NPAR(9)
C
      DO 100 N=1,NEL1
      NEL   = N
      NODS  = IELT(N)
      MTYPE = MATP(N)
      NNOD5 = NODS - 4
      NDOF  = NODS*NODS
C
C     ZERO ELEMENT CONDUCTIVITY MATRIX SK(NODS,NODS)
C
      DO 10 I=1,NDOF
   10 SK(I) = 0.0
C
C     ZERO ELEMENT HF(NODS) VECTOR -
C     INFLUENCE COEFFICIENTS FOR UNIT INTERNAL HEAT GENERATION
C
      IF(IHFLG.EQ.0) GO TO 30
      DO 20 I=1,NODS
   20 HF(I) = 0.0
C
C     ZERO ELEMENT HEAT CAPACITY VECTOR SC(NODS)
C
   30 IF(KST.EQ.-1) GO TO 50
      DO 40 I=1,NODS
   40 SC(I) = 0.0
C
C     DETERMINE ELEMENT NODAL TEMPERATURE VECTOR
C
   50 DO 60 I=1,NODS
      ND = LM(I,N)
   60 TNOD(I) = T(ND)
C
C     CALCULATE ELEMENT CONDUCTIVITY MATRIX USING GAUSS QUADRATURE
C
      CALL FORMK1 (SK,SC,HF,XY(1,N),NOD5(1,N),TNOD,NTC(MTYPE),
     1             DENS(MTYPE),TPROP(1,1,MTYPE),MAXTP,NUMAT,NODS)
C
      IF(IHFLG.EQ.0) GO TO 70
      WRITE(2) (HF(I),I=1,NODS)
C
C     ASSEMBLE EFFECTIVE ELEMENT CONDUCTIVITY MATRIX (SK*)
C     INTO EFFECTIVE STRUCTURAL CONDUCTIVITY MATRIX (K*)
C
   70 CALL ADDBAN (A(N10),A(N9),SK,LM(1,N),NODS)
C
      IF(KST.EQ.-1) GO TO 100
C
C     ASSEMBLE EFFECTIVE LINEAR HEAT CAPACITY VECTOR (C*)
C                         C* = (1./DT)*C
C
      CALL ADDC (A(N12),SC,LM(1,N),NODS)
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FORMK1 (SK,SC,HF,XY,NOD5,TNOD,NTP,DE,TPROP,MAXTP,
     1                   NUMAT,NODS)
C
C     ****************************************************************
C     ISOPARAMETRIC FORMULATION OF ELEMENT CONDUCTIVITY AND HEAT
C     CAPACITY FOR TWO DIMENSIONAL PLANAR AND AXISYMMETRIC GEOMETRY
C     ****************************************************************
C
      DIMENSION  SK(NODS,NODS),SC(NODS),HF(1),XY(1),NOD5(1),TNOD(1),
     1           TPROP(MAXTP,5)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /WORK  / DUM(98),H(8),P(2,8),B(2,8),XJ(2,2),EK(3),WORK(55)
      DIMENSION  XG(4,4),WGT(4,4)
      DATA XG /     0.,                0.,                0.,                0.,
     1 -.5773502691896, .5773502691896,                0.,                0.,
     2 -.7745966692415, .0000000000000, .7745966692415,                0.,
     3 -.8611363115941,-.3399810435849, .3399810435849, .8611363115941/
      DATA WGT / 2.000,                0.,                0.,                0.,
     1 1.0000000000000,1.0000000000000,                0.,                0.,
     2  .5555555555556, .8888888888889, .5555555555556,                0.,
     3  .3478548451375, .6521451548625, .6521451548625, .3478548451375/
C
      ITYP2D = NPAR(4)
      NINT   = NPAR(6)
      IHFLG  = NPAR(9)
C
C     LOOP OVER ALL INTEGRATION POINTS
C
      VOL = 0.0
C
      DO 100 LX=1,NINT
      R  = XG(LX,NINT)
      DO 100 LY=1,NINT
      S  = XG(LY,NINT)
      WT = WGT(LX,NINT)*WGT(LY,NINT)
C
C     FIND INTERPOLATION FUNCTIONS (H) AND THEIR DERIVATIVES (P).
C     FIND JACOBIAN (XJ) AND ITS DETERMINANT (DETJ).
C
      CALL SHAPE1 (R,S,XY,H,P,NOD5,XJ,DETJ)
C
C     EVALUATE JACOBIAN INVERSE (XJI) AND GLOBAL DERIVATIVE OPERATOR (B)
C     AT EACH INTEGRATION POINT (R,S) WITHIN THE ELEMENT
C
      CALL DERIV1 (XY,H,P,B,XJ,DETJ,RAD,ITYP2D)
C
      FAC = WT*RAD*DETJ
      VOL = VOL + FAC
```

```
C
      IF(IHFLG.EQ.0) GO TO 8
      DO 5 I=1,NODS
    5 HF(I) = HF(I) + H(I)*FAC
C
C     COMPUTE DIAGONAL TERMS FOR CONSISTENT HEAT CAPACITY MATRIX
C
    8 IF(KST.EQ.-1) GO TO 20
      DO 10 I=1,NODS
   10 SC(I) = SC(I) + H(I)*H(I)*FAC
C
C     COMPUTE TEMPERATURE AT INTEGRATION POINT (R,S)
C
   20 IF(NTP.EQ.1) GO TO 40
      TIP = 0.
      DO 30 I=1,NODS
   30 TIP = TIP + H(I)*TNOD(I)
C
C     BASED ON TEMPERATURE AT INTEGRATION POINT (TIP), COMPUTE THE
C     TEMPERATURE DEPENDENT CONDUCTIVITIES AND SPECIFIC HEAT
C
   40 CALL MATPR (TIP,EK,SP,TPROP(1,1),TPROP(1,2),NTP,MAXTP,KST)
C
      EK11 = EK(1)
      EK22 = EK(2)
      EK12 = EK(3)
C
C     FORM  SK = B(TRANSPOSE)*EK*B FOR INTEGRATION POINT (R,S)
C
      DO 50 I=1,NODS
      BT1 = B(1,I)
      BT2 = B(2,I)
      DO 50 J=I,NODS
      DB1 = EK11*B(1,J) + EK12*B(2,J)
      DB2 = EK12*B(1,J) + EK22*B(2,J)
   50 SK(I,J) = SK(I,J) + (BT1*DB1 + BT2*DB2)*FAC
C
  100 CONTINUE
C
      NODM = NODS - 1
      DO 200 I=1,NODM
      II = I + 1
      DO 200 J=II,NODS
  200 SK(J,I) = SK(I,J)
C
      IF(KST.EQ.-1) GO TO 600
C
      AO = 1./DT
C
C     COMPUTE SUM OF DIAGONAL HEAT CAPACITY TERMS
C
      SUM = 0.0
      DO 300 I=1,NODS
  300 SUM = SUM + SC(I)
C
C     CORRECTION FACTOR
C
      CF = VOL/SUM
C
C     APPLY CORRECTIONS TO TERMS SC(I)
C
      DO 400 I=1,NODS
  400 SC(I) = DE*SP*AO*CF*SC(I)
C
C     FORM EFFECTIVE ELEMENT CONDUCTIVITY MATRIX (SK*)
C                          SK* = SK + SC*
C
      DO 500 I=1,NODS
  500 SK(I,I) = SK(I,I) + SC(I)
C
  600 RETURN
      END
```

```
      SUBROUTINE SHAPE1 (R,S,XY,H,P,NOD5,XJ,DETJ)
C
C     ******************************************************************
C     1. TO FIND INTERPOLATION FUNCTIONS ( H )
C        AND DERIVATIVES ( P ) CORRESPONDING TO THE NODAL POINTS
C        OF A 4- TO 8-NODE TWO DIMENSIONAL ISOPARAMETRIC ELEMENT
C     2. TO FIND JACOBIAN ( XJ ) AND ITS DETERMINANT ( DETJ )
C
C
C                         NODE NUMBERING CONVENTION
C
C
C                 2                  5                  1
C
C                  0 . . . . . . . . 0 . . . . . . . . 0
C                  .                                   .
C                  .                                   .
C                  .                 S                 .
C                  .                 .                 .
C                  .                 .                 .
C                6 0                 . . . R           0 8
C                  .                                   .
C                  .                                   .
C                  .                                   .
C                  .                                   .
C                  .                                   .
C                  0 . . . . . . . . 0 . . . . . . . . 0
C
C                 3                  7                  4
C
C     ******************************************************************
C
      DIMENSION  XY(2,1),H(1),P(2,1),NOD5(1),XJ(2,2)
      COMMON /TODIM / NEL,NODS,NTYPE,NND5
      DIMENSION  IPERM(4)
      DATA IPERM/2,3,4,1/
C
C     INTERPOLATION FUNCTIONS (4-NODE ELEMENT)
C
      RP   = 1.0 + R
      SP   = 1.0 + S
      RM   = 1.0 - R
      SM   = 1.0 - S
      R2   = 1.0 - R*R
      S2   = 1.0 - S*S
C
      H(1) = 0.25*RP*SP
      H(2) = 0.25*RM*SP
      H(3) = 0.25*RM*SM
      H(4) = 0.25*RP*SM
C
C     LOCAL DERIVATIVES OF INTERPOLATION FUNCTIONS (4-NODE ELEMENT)
C
      P(1,1) = 0.25*SP
      P(1,2) = -P(1,1)
      P(1,3) = -0.25*SM
      P(1,4) = -P(1,3)
      P(2,1) = 0.25*RP
      P(2,2) = 0.25*RM
      P(2,3) = -P(2,2)
      P(2,4) = -P(2,1)
```

```
C
C     INTERPOLATION FUNCTIONS AND LOCAL DERIVATIVES FOR MIDSIDE NODES
C
      IF(NODS.EQ.4) GO TO 50
C
      I = 0
    2 I = I + 1
      IF (I.GT.NNDS) GO TO 40
      NN = NODS(I) - 4
      GO TO (5,6,7,8) NN
C
    5 H(5)   = 0.50*R2*SP
      P(1,5) = -R*SP
      P(2,5) = 0.50*R2
      GO TO 2
    6 H(6)   = 0.50*RM*S2
      P(1,6) = -0.50*S2
      P(2,6) = -RM*S
      GO TO 2
    7 H(7)   = 0.50*R2*SM
      P(1,7) = -R*SM
      P(2,7) = -0.50*R2
      GO TO 2
    8 H(8)   = 0.50*RP*S2
      P(1,8) = 0.50*S2
      P(2,8) = -RP*S
      GO TO 2
C
C     MODIFY INTERPOLATION FUNCTIONS H(1) TO H(4) AND LOCAL DERIVATIVES
C
   40 IH = 0
   41 IH = IH + 1
      IF(IH.GT.NNDS) GO TO 50
      IN = NODS(IH)
      I1 = IN - 4
      I2 = IPERM(I1)
      H(I1) = H(I1) - 0.5*H(IN)
      H(I2) = H(I2) - 0.5*H(IN)
      H(IH+4) = H(IN)
      DO 45 J=1,2
      P(J,I1) = P(J,I1) - 0.5*P(J,IN)
      P(J,I2) = P(J,I2) - 0.5*P(J,IN)
   45 P(J,IH+4) = P(J,IN)
      GO TO 41
C
C     EVALUATE THE JACOBIAN MATRIX AT POINT (R,S)
C
   50 DO 100 I=1,2
      DO 100 J=1,2
      SUM = 0.0
      DO 90 K=1,NODS
   90 SUM = SUM + P(I,K)* XY(J,K)
  100 XJ(I,J) = SUM
C
C     COMPUTE THE DETERMINANT OF THE JACOBIAN MATRIX AT POINT (R,S)
C
      DETJ = XJ(1,1)* XJ(2,2) - XJ(2,1)* XJ(1,2)
      DUM = ABS(DETJ)
      IF(DUM.GT.1.0E-8) GO TO 500
      WRITE (6,3000) NEL
      STOP
C
 3000 FORMAT(//49H **ERROR**  ZERO JACOBIAN DETERMINANT FOR ELEMENT,I5)
C
  500 RETURN
      END
```

```
      SUBROUTINE DERIV1 (XY,H,P,B,XJ,DETJ,RAD,ITYP2D)
C
C     ******************************************************************
C     EVALUATION OF THE GLOBAL DERIVATIVE OPERATOR (B) AT A POINT (R,S)
C     FOR A QUADRILATERAL ELEMENT HAVING PLANAR OR AXISYMMETRIC GEOMETRY
C     ******************************************************************
C
      DIMENSION  XY(2,1),H(1),P(2,1),B(2,1),XJ(2,2)
      COMMON /TODIM / NEL,NODS,MTYPE,NNDS
      COMMON /WORK  / DUM(145),XJI(2,2),WORK(51)
C
C     COMPUTE INVERSE OF THE JACOBIAN MATRIX
C
      DETJI    = 1.0/DETJ
      XJI(1,1) =  XJ(2,2)* DETJI
      XJI(1,2) = -XJ(1,2)* DETJI
      XJI(2,1) = -XJ(2,1)* DETJI
      XJI(2,2) =  XJ(1,1)* DETJI
C
C     EVALUATE GLOBAL DERIVATIVE OPERATOR ( B-MATRIX )
C
      DO 10 K=1,NODS
      B(1,K) = XJI(1,1)*P(1,K) + XJI(1,2)*P(2,K)
   10 B(2,K) = XJI(2,1)*P(1,K) + XJI(2,2)*P(2,K)
C
      RAD = 1.0
      IF(ITYP2D.NE.0) GO TO 500
C
C     COMPUTE THE RADIUS AT POINT (R,S) FOR AXISYMMETRIC SOLID
C
      RAD = 0.0
      DO 50 K=1,NODS
   50 RAD = RAD + H(K)* XY(1,K)
C
      IF(RAD.GT.1.0E-8) GO TO 500
      WRITE(6,3000) NEL
      STOP
C
 3000 FORMAT(//50H **ERROR**  ZERO RADIUS ENCOUNTERED IN ELEMENT NO.,I5)
C
  500 RETURN
      END
```

```
      SUBROUTINE MATPR (TEM,EK,SP,T,PROP,NTP,MAXTP,KST)
C
C     ******************************************************************
C     COMPUTE THE TEMPERATURE DEPENDENT PROPERTIES AS A FUNCTION
C     OF THE KNOWN TEMPERATURE (TEM) AND MATERIAL TYPE, ASSUMING A
C     LINEAR VARIATION BETWEEN TEMPERATURES (T) LISTED IN THE TABLE
C     AND THE CORRESPONDING PROPERTIES (PROP) FOR EACH MATERIAL
C
C                 KST = -1 : K11(T),K22(T),K12(T)
C                 KST =  0 : K11(T),K22(T),K12(T),C(T)
C     ******************************************************************
C
      DIMENSION  EK(3),T(MAXTP),PROP(MAXTP,4)
C
      IF(NTP.GT.1) GO TO 20
C
C     MATERIAL PROPERTIES CONSTANT WITH TEMPERATURE
C
   10 EK(1) = PROP(1,1)
      EK(2) = PROP(1,2)
      EK(3) = PROP(1,3)
      IF(KST.EQ.-1) GO TO 100
      SP     = PROP(1,4)
      GO TO 100
C
C     CHECK TEMPERATURE BOUNDS
C
   20 IF(TEM.GE.T(1)) GO TO 30
      WRITE(6,3000) TEM
      STOP
   30 IF(TEM.LE.T(NTP)) GO TO 40
      WRITE(6,3000) TEM
      STOP
   40 IF(TEM.EQ.T(1)) GO TO 10
C
      DO 50 J=2,NTP
      N  = J
      NM = J - 1
      IF(TEM.GT.T(NM).AND.TEM.LE.T(N)) GO TO 60
   50 CONTINUE
C
   60 DTEMP = T(N) - T(NM)
      RATIO = (TEM-T(NM))/DTEMP
C
      DO 70 I=1,3
   70 EK(I) = PROP(NM,I) + RATIO*(PROP(N,I)-PROP(N4,I))
      IF(KST.EQ.-1) GO TO 100
      SP     = PROP(NM,4) + RATIO*(PROP(N,4)-PROP(NM,4))
C
 3000 FORMAT(//50H **ERROR**   COMPUTED TEMPERATURE OUT OF RANGE FOR,
     1        40H VALUES GIVEN IN MATERIAL TABLE .... T =,E10.3)
C
  100 RETURN
      END
```

```
      SUBROUTINE ADDBAN (A,MAXA,S,LM,NDOF)
C
C     ******************************************************************
C     ASSEMBLES ELEMENT CONDUCTIVITY INTO COMPACTED GLOBAL CONDUCTIVITY
C     ******************************************************************
C
      DIMENSION  A(1),MAXA(1),S(1),LM(1)
C
      DO 200 J=1,NDOF
      JJ = LM(J)
      MJ = MAXA(JJ)
      DO 200 I=1,NDOF
      II = LM(I)
      IJ = JJ - II
      IF(IJ) 200,100,100
  100 KK = MJ + IJ
      LS = (J-1)*NDOF + I
      A(KK) = A(KK) + S(LS)
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE ADDC (A,SC,LM,IEL)
C
C     ******************************************************************
C     ADD ELEMENT HEAT CAPACITY TO GLOBAL LUMPED HEAT CAPACITY VECTOR
C     ******************************************************************
C
      DIMENSION  A(1),SC(1),LM(1)
C
      DO 100 I=1,IEL
      II=LM(I)
  100 A(II) = A(II) + SC(I)
C
      RETURN
      END
```

```
      SUBROUTINE COND2 (LM,XX,CL,MATP,NFH,PROP,NUMAT)
C
C     ******************************************************************
C     MODIFY SYSTEM CONDUCTIVITY MATRIX (K*) FOR LINEAR CONVECTION BC
C     ******************************************************************
C
      DIMENSION  LM(2,1),XX(2,1),CL(1),MATP(1),NFH(1),PROP(5,NUMAT)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /WORK  / DUM(10),HK(2,2),WORK(186)
      COMMON  A(1)
C
      NEL2 = NPAR(2)
      ITYP = NPAR(4)
C
      DO 100 N=1,NEL2
      MT = MATP(N)
      NF = NFH(MT)
      IF(NF.NE.0) GO TO 100
      H  = PROP(1,MT)
      HL = H*CL(N)
C
C     FORM HK(2,2) FOR EACH LINEAR CONVECTION SURFACE ELEMENT
C
      IF(ITYP.EQ.0) GO TO 30
C
C     PLANAR SOLID CONVECTION BOUNDARY ELEMENTS
C
      HK(1,1) = HL/3.
      HK(1,2) = HL/6.
      GO TO 50
C
C     AXISYMMETRIC SOLID CONVECTION BOUNDARY ELEMENTS
C
   30 XI      = XX(1,N)
      XJ      = XX(2,N)
      HK(1,1) = HL*(XI + XJ/3.)/4.
      HK(1,2) = HL*(XI + XJ)/12.
C
   50 HK(2,2) = HK(1,1)
      HK(2,1) = HK(1,2)
C
C     ASSEMBLE ELEMENT CONDUCTIVITY MATRIX INTO STRUCTURAL CONDUCTIVITY
C
      CALL ADDBAN (A(N10),A(N9),HK,LM(1,N),2)
C
  100 CONTINUE
C
```

```
      RETURN
      END
```

```
      SUBROUTINE COND3 (LM,H,T3,TD)
C
C     **********************************************************************
C     MODIFY SYSTEM CONDUCTIVITY MATRIX (K*) FOR COOLING PIPE EFFECTS
C     **********************************************************************
C
      DIMENSION  LM(1),H(1),TB(1),TD(1)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON  A(1)
C
      NEL3 = NPAR(2)
C
      DO 100 N=1,NEL3
      IF(TB(N).GT.TIME) GO TO 100
      IF(TD(N).LE.TIME) GO TO 100
C
      CALL ADDBAN (A(N10),A(N9),H(N),LM(N),1)
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE MODKTB (NOD,KODE,MAXA,XK,NNBC)
C
C     **********************************************************************
C     MODIFY CONDUCTIVITY MATRIX FOR TEMPERATURE BOUNDARY CONDITIONS
C     **********************************************************************
C
      DIMENSION  NOD(1),KODE(1),MAXA(1),XK(1)
C
      DO 100 N=1,NNBC
      IF(KODE(N).EQ.0) GO TO 100
      NND=NOD(N)
      NDI=MAXA(NND)
      XK(NDI) = XK(NDI) + 1.0E+10
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FORME (T,E,C,NUMNP)
C
C     **********************************************************************
C     FORM THE THERMAL RESISTANCE VECTOR (E)
C     **********************************************************************
C
      DIMENSION T(1),E(1),C(1)
C
      DO 10 I=1,NUMNP
   10 E(I) = C(I)*T(I)
C
      RETURN
      END
```

```
      SUBROUTINE COLSOL (A,V,MAXA,NN,MA,NWA,KKK)
C
C     **********************************************************************
C     TO SOLVE SIMULTANEOUS EQUATIONS AX=V IN CORE, USING
C     COMPACTED STORAGE AND COLUMN REDUCTION SCHEME.
C
C     A    = MATRIX STORED IN COMPACTED FORM
C     V    = VECTOR TO BE REDUCED
C     MAXA = VECTOR CONTAINING ADDRESSES OF DIAGONAL ELEMENTS OF A
C
C     FLAG FOR TRIANGULARIZATION (A=LU) AND/OR SIMPLE FORWARD
C     REDUCTION (LY=V) AND BACKSUBSTITUTION (UX=Y):
C
C        KKK=0  TRIANGULARIZATION ONLY
C        KKK=1  TRIANGULARIZATION PLUS SOLUTION
C        KKK=2  FORWARD REDUCTION AND BACKSUBSTITUTICN ONLY
C        KKK=3  BACKSUBSTITUTION ONLY
C     **********************************************************************
C
      DIMENSION A(NWA),V(1),MAXA(1)
C
      MA1=MA - 1
      IF (KKK-2) 100,700,800
C
C     =================
C     TRIANGULARIZATION
C     =================
C
  100 IF(NN.EQ.1) GO TO 800
      N=1
      IF (A(1)) 80,85,110
   80 WRITE (6,3000) N
      STOP
   85 WRITE (6,3001) N
      STOP
C
  110 DO 200 N=2,NN
      KL=MAXA(N) + 1
      KU=MAXA(N+1) - 1
      IF (KU-KL) 200,210,210
  210 B=0.
C
      KN=MAXA(N)
      K=N
      DO 220 KK=KL,KU
      K=K - 1
      KI=MAXA(K)
      C=A(KK)/A(KI)
      B=B + C*A(KK)
  220 A(KK)=C
      A(KN)=A(KN) - B
C
      IF (A(KN)) 222,224,226
  222 WRITE (6,3000) N
      STOP
  224 WRITE (6,3001) N
```

```
      STOP
  226 MR=MINO(MA1,NN-N)
      IF (MR) 200,200,228
  228 MN=KU - KL + 1
C
      DO 240 J=1,MR
      MJ=MAXA(N+J) + J
      MNJ=MAXA(N+J+1) - MJ - 1
      IF (MNJ) 240,240,230
  230 ND=MINO(MN,MNJ)
      C=0.
      KU=KN + ND
      IC=MJ - KN
      DO 300 KK=KL,KU
  300 C=C + A(KK)*A(KK+IC)
      A(KN+IC)=A(KN+IC) - C
  240 CONTINUE
C
  200 CONTINUE
      IF(KKK.EQ.0) RETURN
C
C     =================
C     FORWARD REDUCTION
C     =================
C
  700 DO 400 N=2,NN
      KL=MAXA(N) + 1
      KU=MAXA(N+1) - 1
      IF (KU-KL) 400,410,410
  410 K=N
      C=0.
      DO 420 KK=KL,KU
      K=K - 1
  420 C=C + A(KK)*V(K)
      V(N)=V(N) - C
  400 CONTINUE
      GO TO 800
C
C     =================
C     BACK SUBSTITUTION
C     =================
C
  800 DO 480 N=1,NN
      K=MAXA(N)
  480 V(N)=V(N)/A(K)
      IF (NN.EQ.1) RETURN
      N=NN
      DO 500 L=2,NN
      KL=MAXA(N) + 1
      KU=MAXA(N+1) - 1
      IF (KU-KL) 500,510,510
  510 K=N
      DO 520 KK=KL,KU
      K=K - 1
  520 V(K)=V(K) - A(KK)*V(N)
  500 N=N - 1
C
C     FORMAT STATEMENTS
C
 3000 FORMAT(//45H **STOP**  STIFFNESS NOT POSITIVE DEFINITE.../,
     1        10X,27H NEGATIVE PIVOT IN POSITION I4)
 3001 FORMAT(//33H **STOP**  ZERO PIVOT IN POSITION I4)
C
      RETURN
      END
```

```
      SUBROUTINE FORMQP (TFN,FN,NPTS,NOD,KODE,NFN,TQ,Q,NPTM1)
C
C     ******************************************************************
C     THIS ROUTINE FORMS THE PRESCRIBED FLUX VECTOR (Q) AT TIME (TIME)
C     ******************************************************************
C
      DIMENSION  TFN(NPTM1,1),FN(NPTM1,1),NPTS(1),NOD(1),KODE(1),
     1           NFN(1),TQ(1),Q(1)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /NBC   / NNBC,NBCF,NPTM
C
C     INITIALIZE FLUX VECTOR (Q)
C
      DO 10 I=1,NUMNP
   10 Q(I) = 0.0
C
      DO 50 N=1,NNBC
      II  = NOD(N)
      NF  = NFN(N)
      IF(NF.EQ.0) GO TO 20
      NPT = NPTS(NF)
      CALL INTERP (TFN(1,NF),FN(1,NF),NPT,TIME,VAL)
   20 IF(KODE(N).EQ.0) GO TO 40
C
C     MODIFY FLUX VECTOR FOR TEMPERATURE BOUNDARY CONDITIONS
C
      IF(NF.NE.0) GO TO 30
      Q(II) = Q(II) + (1.0E+10)*TQ(N)
      GO TO 50
   30 Q(II) = Q(II) + (1.0E+10)*TQ(N)*VAL
      GO TO 50
C
C     MODIFY FLUX VECTOR FOR PRESCRIBED FLUXES
C
   40 IF(NF.EQ.0) VAL = 1.0
      Q(II) = Q(II) + TQ(N)*VAL
C
   50 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FORMQC
C
C     ******************************************************************
C     THIS ROUTINE MODIFIES THE FLUX VECTOR (Q) FOR HEAT GENERATION,
C     CONVECTION AND RADIATION BOUNDARIES AND COOLING PIPES
C     ******************************************************************
C
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /NBC   / NNBC,NBCF,NPTM
      COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
      DIMENSION  NST(10)
      EQUIVALENCE  (NST(1),M1)
C
      REWIND 1
      REWIND 2
C
C     LOOP OVER ALL ELEMENT GROUPS
C
```

```
      DO 100 NG=1,NEG
C
      READ (1) MIDEST,NPAR,NST,(A(I),I=1,MIDEST)
C
      NGR = NPAR(1)
C
      GO TO (1,2,3) NGR
C
    1 IHFLG  = NPAR(9)
      IF(IHFLG.EQ.0) GO TO 100
      MXNODS = NPAR(5)
      CALL FLUX1 (A(M1),A(M3),A(M5),A(M9),A(M10),A(N1),A(N2),A(N3),
     1            A(N11),MXNODS,NPTM)
      GO TO 100
C
    2 NUMAT = NPAR(5)
      CALL FLUX2 (A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),A(M7),A(M8),A(M9),
     1            A(N1),A(N2),A(N3),A(N8),A(N11),A(N14),NUMAT,NPTM)
      GO TO 100
C
    3 CALL FLUX3 (A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),A(N1),A(N2),
     1            A(N3),A(N11),NPTM)
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FLUX1 (LM,IELT,MATP,NFHG,HGSC,TFN,FN,NPTS,Q,
     1                  MXNODS,NPTM)
C
C     *****************************************************************
C     THIS ROUTINE MODIFIES FLUX VECTOR (Q) FOR HEAT GENERATION
C     *****************************************************************
C
      DIMENSION  LM(MXNODS,1),IELT(1),MATP(1),NFHG(1),HGSC(1),
     1           TFN(NPTM,1),FN(NPTM,1),NPTS(1),Q(1)
C
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
      COMMON /WORK  / HF(8),WORK(192)
C
      NEL1 = NPAR(2)
C
      DO 100 N=1,NEL1
      NODS = IELT(N)
      MTYP = MATP(N)
      NFH  = NFHG(MTYP)
      HGS  = HGSC(MTYP)
C
      READ (2) (HF(I),I=1,NODS)
C
C     COMPUTE RATE OF INTERNAL HEAT GENERATION
C
      VAL = 1.0
      IF(NFH.EQ.0) GO TO 20
      NPT = NPTS(NFH)
      CALL INTERP (TFN(1,NFH),FN(1,NFH),NPT,TIME,VAL)
   20 QG = HGS*VAL
C
      DO 30 I=1,NODS
   30 HF(I) = HF(I)*QG
C
C     ASSEMBLE ELEMENT HEAT FLUX VECTOR INTO STRUCTURAL VECTOR (Q)
C
      CALL ADDC (Q,HF,LM(1,N),NODS)
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FLUX2 (LM,XX,CL,MATP,SB,TSHIFT,NFH,NFTE,PROP,TFN,FN,
     1                  NPTS,T,Q,TT,NUMAT,NPTM)
C
C     ********************************************************************
C     MODIFY FLUX VECTOR (Q) FOR CONVECTION AND RADIATION BOUNDARIES
C     ********************************************************************
C
      DIMENSION  LM(2,1),XX(2,1),CL(1),MATP(1),NFH(1),NFTE(1),T(1),
     1           PROP(5,NUMAT),TFN(NPTM,1),FN(NPTM,1),NPTS(1),Q(1),TT(1)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
C
      NEL2 = NPAR(2)
      ITYP = NPAR(4)
      IF(KBC.EQ.0) GO TO 10
C
C     ESTIMATE TEMPERATURES AT MIDPOINT OF NEXT TIME STEP
C
      DO 5 I=1,NUMNP
    5 TT(I) = T(I) + 0.5*(T(I)-TT(I))
C
   10 DO 100 N=1,NEL2
      II   = LM(1,N)
      JJ   = LM(2,N)
      XL   = CL(N)
      MTYP = MATP(N)
      NFS  = NFH(MTYP)
      NFT  = NFTE(MTYP)
      HAM  = PROP(1,MTYP)
      TEX  = PROP(2,MTYP)
      V    = PROP(3,MTYP)
C
C     COMPUTE EXTERNAL TEMPERATURE
C
      IF(NFT.NE.0) GO TO 15
      VAL = 1.0
      GO TO 20
   15 NPT = NPTS(NFT)
      CALL INTERP (TFN(1,NFT),FN(1,NFT),NPT,TIME,VAL)
   20 TE  = VAL*TEX
C
C     ============================
C     CONVECTION BOUNDARY CONDITIONS
C     ============================
C
      IF(NFS.GT.0) GO TO 30
C
C     LINEAR CONVECTION, H = CONSTANT
C
      HL = HAM*XL
      IF(ITYP.EQ.0) GO TO 25
C
C     PLANAR SOLID CONVECTION BOUNDARY ELEMENTS
C
      QC = HL*TE/2.
      Q(II) = Q(II) + QC
      Q(JJ) = Q(JJ) + QC
      GO TO 35
C
C     AXISYMMETRIC SOLID CONVECTION BOUNDARY ELEMENTS
C
   25 XI = XX(1,N)
      XJ = XX(2,N)
      QC = HL*TE/6.
      Q(II) = Q(II) + QC*(2.*XI + XJ)
      Q(JJ) = Q(JJ) + QC*(2.*XJ + XI)
      GO TO 35
C
C     NONLINEAR CONVECTION, H = H(T)
C
   30 TS   = 0.5*(TT(II) + TT(JJ))
      TAVE = 0.5*(TE + TS)
C
C     COMPUTE TEMPERATURE DEPENDENT CONVECTION COEFFICIENT
C
      NTP = NPTS(NFS)
      CALL HPROP (TAVE,H,TFN(1,NFS),FN(1,NFS),NTP)
      QC = 0.5*HAM*H*XL*(TE - TS)
      Q(II) = Q(II) + QC
      Q(JJ) = Q(JJ) + QC
C
C     ============================
C     RADIATION BOUNDARY CONDITIONS
C     ============================
C
   35 IF(V.EQ.0.) GO TO 100
      TS = 0.5*(TT(II) + TT(JJ)) + TSHIFT
      TS = TS*TS
      TS = TS*TS
      TE = TE + TSHIFT
      TE = TE*TE
      TE = TE*TE
      EF = PROP(4,MTYP)
      ES = PROP(5,MTYP)
      QR = 0.5*SB*V*XL*(1./(1./EF + 1./ES - 1.))*(TE - TS)
      Q(II) = Q(II) + QR
      Q(JJ) = Q(JJ) + QR
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE HPROP (TAVE,H,T,HT,NTP)
C
C     ******************************************************************
C     COMPUTE TEMPERATURE DEPENDENT CONVECTION COEFFICIENTS H(T)
C     ******************************************************************
C
      DIMENSION  T(NTP),HT(NTP)
C
C     CHECK TEMPERATURE BOUNDS
C
      IF(TAVE.GE.T(1)) GO TO 10
      WRITE(6,3000) TAVE
      STOP
   10 IF(TAVE.LE.T(NTP)) GO TO 20
      WRITE(6,3000) TAVE
      STOP
C
   20 IF(TAVE.NE.T(1)) GO TO 30
      H = HT(1)
      GO TO 100
C
   30 DO 50 J=2,NTP
      N  = J
      NM = J - 1
      IF(TAVE.GT.T(NM).AND.TAVE.LE.T(N)) GO TO 60
   50 CONTINUE
C
   60 DTEMP = T(N) - T(NM)
      RATIO = (TAVE-T(NM))/DTEMP
      H = HT(NM) + RATIO*(HT(N)-HT(NM))
C
 3000 FORMAT(//50H **ERROR**   COMPUTED TEMPERATURE OUT OF RANGE FOR,
     1       42H VALUES GIVEN IN CONVECTION TABLE .... T = E10.3)
C
  100 RETURN
      END
```

```
      SUBROUTINE FLUX3 (LM,H,NFCP,TW,TB,TD,TFN,FN,NPTS,Q,NPTM)
C
C     ******************************************************************
C     THIS ROUTINE MODIFIES FLUX VECTOR (Q) FOR COOLING PIPES
C     ******************************************************************
C
      DIMENSION  LM(1),H(1),NFCP(1),TW(1),TB(1),TD(1),TFN(NPTM,1),
     1           FN(NPTM,1),NPTS(1),Q(1)
      COMMON /CNTRL1/ NUMNP,NEG,MODEX,NPAR(10),NG,KBC
      COMMON /CNTRL2/ KST,NDT,DT,TSTART,TAMB,NPRINT,NTSREF,TIME,KP
C
      NEL3 = NPAR(2)
C
      DO 100 N=1,NEL3
      IF(TB(N).GT.TIME) GO TO 100
      IF(TD(N).LE.TIME) GO TO 100
      II = LM(N)
      NF = NFCP(N)
C
      IF(NF.EQ.0) GO TO 20
      NPT = NPTS(NF)
      CALL INTERP (TFN(1,NF),FN(1,NF),NPT,TIME,VAL)
C
   20 IF(NF.EQ.0) VAL = 1.0
      Q(II) = Q(II) + H(N)*TW(N)*VAL
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INTERP (TFN,FN,NPT,TIME,VAL)
C
C     ******************************************************************
C     THIS ROUTINE INTERPOLATES A GIVEN TIME-DEPENDENT FUNCTION TO FIND
C     THE VALUE OF THE FUNCTION (VAL) AT A PARTICULAR TIME POINT (TIME)
C     ******************************************************************
C
```

```
      DIMENSION  TFN(1),FN(1)
C
      DO 10 N=1,NPT
      DTIME = TFN(N) - TIME
      IF(DTIME.GT.0.) GO TO 15
   10 CONTINUE
C
   15 DIFF = TFN(N) - TFN(N-1)
      VAL  = FN(N) - (FN(N) - FN(N-1))*DTIME/DIFF
C
      RETURN
      END
```

```
      SUBROUTINE QEFF (Q,E,NUMNP)
C
C     ******************************************************************
C     FORM EFFECTIVE HEAT FLUX VECTOR (Q*)
C     ******************************************************************
C
      DIMENSION Q(1),E(1)
C
      DO 10 I=1,NUMNP
   10 Q(I) = Q(I) + E(I)
C
      RETURN
      END
```

```
      SUBROUTINE PNODE (X,Y,NUMNP)
C
C     ******************************************************************
C     PUNCH THE NODAL POINT COORDINATE DATA
C     ******************************************************************
C
      DIMENSION  X(1),Y(1)
C
      WRITE(3,2000)
      NCARD = NUMNP/3 + 1
      NC = 0
C
      DO 100 I=1,NUMNP,3
      NC = NC + 1
      IP = I + 2
      IF(NC.EQ.NCARD) IP = NUMNP
  100 WRITE(3,2001) (N,X(N),Y(N),N=I,IP)
C
 2000 FORMAT(24H NODAL POINT COORDINATES)
 2001 FORMAT(3(I5,2F10.3))
C
      RETURN
      END
```

```
      SUBROUTINE PTEMP (T,TIME,NUMNP)
C
C     ******************************************************************
C     PUNCH THE NODAL POINT TEMPERATURES
C     ******************************************************************
C
      DIMENSION  T(1)
C
      WRITE(3,2000) TIME
      NCARD = NUMNP/4 + 1
      NC = 0
C
      DO 100 I=1,NUMNP,4
      NC = NC + 1
      IP = I + 3
      IF(NC.EQ.NCARD) IP = NUMNP
  100 WRITE(3,2001) (N,T(N),N=I,IP)
C
 2000 FORMAT(35H NODAL POINT TEMPERATURES AT TIME = F11.4)
 2001 FORMAT(4(I5,5X,F10.3))
C
      RETURN
      END
```

```
      SUBROUTINE OUT (T,NUMNP,TIME,KSTEP)
C
C     ******************************************************************
C     PRINT NODAL TEMPERATURES FOR *TIME*
C     ******************************************************************
C
      DIMENSION  T(1)
C
      WRITE(6,2000) KSTEP,TIME
      WRITE(6,2001) (N,T(N),N=1,NUMNP)
C
C     FORMAT STATEMENTS
C
 2000 FORMAT(3/28H TEMPERATURES AT TIME STEP =I5,2X,7H(TIME =E11.4,1H)/)
 2001 FORMAT(6(I6,E14.6))
C
      RETURN
      END
```

APPENDIX B1

DETECT USER'S MANUAL

PROGRAM IDENTIFICATION

DETECT: Determination of Temperatures in Construction. A Two Dimensional Heat Transfer Program for Structures of Arbitrary Geometry Constructed Incrementally. Version I, Febuary 1976.

Developed by: R. M. Polivka and E. L. Wilson
University of California, Berkeley

CONTENTS

B1.1 PROGRAM DESCRIPTION

The computer program DETECT performs the steady state or transient heat transfer analysis of two dimensional planar or axisymmetric structures of arbitrary geometry constructed incrementally. The program was developed primarily for the purpose of evaluating the temperature distribution during the construction of mass concrete structures such as dams, foundations and bridge piers. The solution is based on a finite element technique, and includes the effects of such time-dependent boundary conditions as convective surface heat transfer (insulation forms), nodal temperature or heat flux specifications, internal heat generation and artificial cooling by means of embedded pipes. Material nonlinearities such as temperature dependent thermal conductivity or heat capacity are neglected.

The program is completely general, and is applicable to structures composed of a combination of element types and materials. The present version contains the following element types:

1. Two dimensional 4- to 8-node planar or axisymmetric solid elements.
2. Convection boundary elements.
3. Cooling pipe elements.

B1.2 PROGRAM CAPACITY

The computer program DETECT uses variable dimensioning in order to make optimum use of available high speed storage. The element data for each group of elements is stored in a block and transferred to disc storage. The required core storage remaining is dynamically allocated into a single array A in blank common.

The program capacity can be controlled by the user through the two fortran statements in the main program of DETECT:

```
COMMON (n)
MTOT = n
```

The total memory n required can be changed depending on the size of the problem to be solved. The minimum value of n needed is given by

$$n = M_{max} + 2*NPTM*NBCF + NBCF + 4*NNBC + 8*NUMNP + NWK + 1$$

where

M_{max} = Maximum storage required for element group data

NPTM = Maximum number of time points used to describe any of the boundary condition functions

NBCF = Number of boundary condition functions

NNBC = Number of nodal point boundary conditions

NUMNP = Number of nodes

NWK = Working storage required for conductivity matrix. Approximate by MB*NUMNP where MB = maximum bandwidth.

M_{max} is the maximum value of M for each of the three element types used in the analysis:

(a) Type 1 - Planar or axisymmetric elements

$$M_1 = 4*MXNODS*NEL1 + 7*NUMAT$$

NEL1 = Number of elements in group 1

NUMAT = Number of material types

MXNODS = Maximum number of nodes defining any element

(b) Type 2 - Convection boundary elements

M_2 = 9*NEL2

NEL2 = Number of elements in group 2

(c) Type 3 - Cooling pipe elements

M_3 = 6*NEL3

NEL3 = Number of elements in group 3

If the value of n is set too small, an error message is printed which gives the amount by which the storage was exceeded and the program execution is terminated.

B1.3 PROGRAM INPUT DATA

The following sections describe the necessary sequence of cards which define a given structure to be analyzed.

I. PROBLEM INITIATION AND TITLE (A5,3X,18A4) - One card

Columns	Variable	Description
1 - 5	MODE	Problem initiation flag. Punch the word START.
6 - 8	--	Blank
9 - 80	HED	Problem title for labeling output.

II. MASTER CONTROL CARD (5I5) - One card

Columns	Variable	Description
1 - 5	NUMNP	Total number of nodal points in structure.
6 - 10	NEG	Number of element groups. See Note 1.
11 - 15	NLT	Total number of layers (time spans) in the structure. Specify (a) Zero or blank: defaults to 1. See Note 2.
16 - 20	NUMEST	Estimated maximum number of high speed storage locations M_{max} required to store each set of element group data. Specify (a) Zero or blank: defaults to 4000.
21 - 25	MODEX	Execution mode. Specify (a) Zero or blank: data check only. (b) 1: execution.

NOTE

(1) An element group is a series of elements of a particular type (e.g. 2/D axisymmetric, 2/D planar, convection etc.). Elements of the same type may also be split into more than one group.

(2) NLT determines the number of card sets to be read in Section VIII.
   (a) For a structure constructed incrementally, NLT represents the number of lifts in the structure.
   (b) For the transient analysis of an already existing structure, NLT represents the number of time span data changes in either the time step increment (DT), the number of solution time steps (NDT) or the output print interval (NPRINT).
   (c) For the steady state analysis of an already existing structure, NLT must equal 1.

III. NODAL POINT COORDINATES (I5,5X,2F10.0,I5,I1)

As many cards as needed to generate NUMNP nodal points.

Columns	Variable	Description
1 - 5	N	Node number. See Note 1.
6 - 10	--	Blank
11 - 20	X(N)	X coordinate
21 - 30	Y(N)	Y coordinate
31 - 35	KN	Node number difference between successive generated nodes (given on first card in a sequence). Specify (a) Zero or blank: No generation. See Note 2.
36	JPR	Print suppression flag (on card for node 1 only). Specify (a) Zero or blank: No suppression. (b) 1: Suppress ordered list of node coordinates. See Note 3.

NOTE

(1) Node cards need not be in numerical order. Eventually, however, all nodes from 1 to the total number of nodes (NUMNP) must be identified.

(2) The mesh generation parameter KN must appear on the first card of a series of nodal points to be generated. The intermediate nodes to be generated between nodes N1 and N2 will be located at equal intervals along the straight line joining the two nodes. KN is the increment to be added to the previous node number. The node difference N2-N1 must be evenly divisible by KN.

(3) JPR is used to eliminate the second printing of the ordered node coordinates. The JPR character is entered on the card for node 1 only.

IV. BOUNDARY CONDITION FUNCTIONS

A. CONTROL INFORMATION (3I5) - One card

Columns	Variable	Description
1 - 5	NBCF	Number of boundary condition functions. See Note 1.
6 - 10	NPTM	Maximum number of points used to describe any one of the functions. See Note 2.
11 - 15	NNBC	Number of nodal point heat flow or temperature boundary conditions. See Note 3.

NOTE

(1) NBCF determines the number of card sets to be read in Section IV.B.

(2) NPTM is the maximum number of [f(t),t] pairs used to define any one of the NBCF functions. At least two points are required to input any one function, and no function may be input with more than NPTM points.

(3) NNBC determines the number of cards to be read or generated in Section V.

B. TIME FUNCTION DATA - NBCF sets of cards

Each set consists of a control card followed by as many cards as needed to define the function.

1. CONTROL CARD (2I5) - First card of set.

Columns	Variable	Description
1 - 5	NC	Function number. (GE.1 and LE.NBCF) See Note 1.
6 - 10	NPTS(NC)	Number of time points used to describe this function. (GE.2 and LE.NPTM)

2. [f(t),t] DATA (8F10.0) - Remaining cards of set.

As many cards as needed to define NPTS(NC) pairs of points [TFN(NC,I),FN(NC,I)], four pairs per card. See Note 2.

Columns	Variable	Description
1 - 10	TFN(NC,1)	Time at point 1 : t_1
11 - 20	FN (NC,1)	Function value at point 1 : $f(t_1)$
21 - 30	TFN(NC,2)	Time at point 2 : t_2
31 - 40	FN (NC,2)	Function value at point 2 : $f(t_2)$
41 - 50	TFN(NC,3)	Point 3
51 - 60	FN (NC,3)	
61 - 70	TFN(NC,4)	Point 4
71 - 80	FN (NC,4)	

NEXT CARD(S) - If required

1 - 10	TFN(NC,5)	Time at point 5 : t_5
11 - 20	FN (NC,5)	Function value at point 5 : $f(t_5)$
. . .	. . .	. . .

NOTE

(1) Time functions need not be input in order of increasing function number NC.

(2) The function tables input in this section are used to prescribe time-dependent boundary conditions such as environmental or nodal point temperatures and nodal point heat flows. The functions may also be used to describe the rate of internal heat generation and the variation of convection coefficient H with temperature.

Time values at successive points must increase in magnitude. Values of the functions at times (temperatures) other than TFN(NC,I) are calculated within the program using linear interpolation.

The first time point TFN(NC,1) must be less than or equal to the time at solution start (PLTIME for layer 1) and the final time point TFN(NC,NPTS(NC)) must be greater than or equal to the time at the end of solution (the sum of NDT*DT for all of the layers).

V. NODAL POINT BOUNDARY CONDITIONS (4I5,F10.0,I5) - NNBC cards

Columns	Variable	Description
1 - 5	N	Boundary condition number. See Note 1.
6 - 10	NOD(N)	Global node number. (GE.1 and LE.NUMNP)
11 - 15	KODE(N)	Boundary condition type. Specify (a) Zero or blank : Externally supplied heat flux Q. (b) 1: Prescribed temperature T.
16 - 20	NFN(N)	Function number. (GE.0 and LE.NBCF) See Note 2.
21 - 30	TQ(N)	Boundary value amplitude. Specify (a) Heat flux (KODE(N) = 0). (b) Temperature (KODE(N) = 1). See Note 3.
31 - 35	KN	Node number difference between successive generated nodes (on first card in sequence). See Note 4.

NOTE

(1) All nodal points not specified in this section are assumed to have externally supplied heat flux of zero for all values of time.

(2) A function number equal to zero or blank means that the prescribed boundary condition is applied at time zero and remains constant for all time greater than zero (step function). The functions assigned in this section must have been defined previously in Section IV. A given function can be used to describe any number of boundary conditions.

(3) (a) For the case of flux or temperature boundary conditions which are not step functions, TQ(N) represents a function multiplier used to scale NFN(N) values for all time t.

(b) For the case of flux or temperature boundary conditions applied as a step function from time zero, TQ(N) represents the actual value of prescribed flux or temperature.

(4) In the printout of nodal point boundary conditions, all generated node data are prefixed by an asterisk. For use of the generation parameter KN, see Note 2, Section III.

VI. ELEMENT SPECIFICATION

Elements must be divided into "groups". Input as many blocks of data in the following sections as there are element groups (NEG). An element group is a series of elements of a particular type, and elements of the same type may also be divided into more than one group.

Element groups may be input in any order. The elements in any group must be numbered sequentially starting from the number of the first element as specified on the element group control card.

The following types of element groups may be used:

Type 1 - Two Dimensional Finite Elements

These are 4- to 8-node isoparametric elements which must be input in the global X-Y plane. When the element is used to represent an axisymmetric solid, the global Y-axis is the axis of revolution.

Type 2 - Convection Boundary Surface Elements

These are 2-node planar or axisymmetric solid boundary elements which must be input in the global X-Y plane.

Type 3 - Cooling Pipe Elements

These are 1-node elements which are placed at existing nodal points in the finite element mesh. The axis of the cooling pipe is the Z-axis.

TYPE 1 - TWO DIMENSIONAL FINITE ELEMENTS

A. CONTROL INFORMATION (8I5) - One card

Columns	Variable	Description
1 - 5	NGR	Element group indicator. Punch the number "1".
6 - 10	NEL1	Number of elements in this group.
11 - 15	MFST	Element number of first element in group. (a) Zero or blank: defaults to 1. See Note 1.
16 - 20	ITYP2D	Element type code. Specify (a) Zero or blank: axisymmetric (b) 1: planar
21 - 25	MXNODS	Maximum number of nodes used to describe any one element. Specify (a) Zero or blank: defaults to 4. (GE.4 and LE.8)
26 - 30	NINT	Numerical integration order to be used in Gaussian quadrature. Specify (a) Zero or blank: defaults to 2. (GE.2 and LE.4) See Note 2.
31 - 35	NUMAT	Number of material property sets. Specify (a) Zero or blank: defaults to 1.
36 - 40	IHFLG	Internal heat generation flag. Specify (a) Zero or blank: No internal heat generation. (b) 1: Internal heat generation exists.

NOTE

(1) Element numbers need not begin with number 1 if MFST is specified.

(2) For rectangular elements, an integration order of 2 is sufficient. If the element is distorted, a higher integration order need be used.

B. MATERIAL PROPERTY INFORMATION (I5,5X,7F10.0) - One card

Columns	Variable	Dimension
1 - 5	M	Material identification number. (GE.1 and LE.NUMAT)
6 - 10	--	Blank
11 - 20	PROP(1)	Thermal conductivity k_{11}.
21 - 30	PROP(2)	Thermal conductivity k_{22}.
31 - 40	PROP(3)	Thermal conductivity k_{12}.
41 - 50	PROP(4)	Thermal heat capacity c.
51 - 60	PROP(5)	Density of material ρ. See Note 1.
61 - 70	PROP(6)	Function number describing the rate of internal heat generation (input as a floating point number). See Note 2.
71 - 80	PROP(7)	Internal heat generation function multiplier.

NOTE

(1) Consistent units must be used for specification of material properties.

(2) If no internal heat generation exists, i.e. if IHFLG equals zero, leave columns 61-80 blank. For the case where internal heat generation is considered, the functions PROP(6) assigned in this section must have been defined previously in Section IV. See Note 2, Section V.

C. ELEMENT DATA (12I5,2F10.0)

As many cards as needed to generate NEL1 elements.

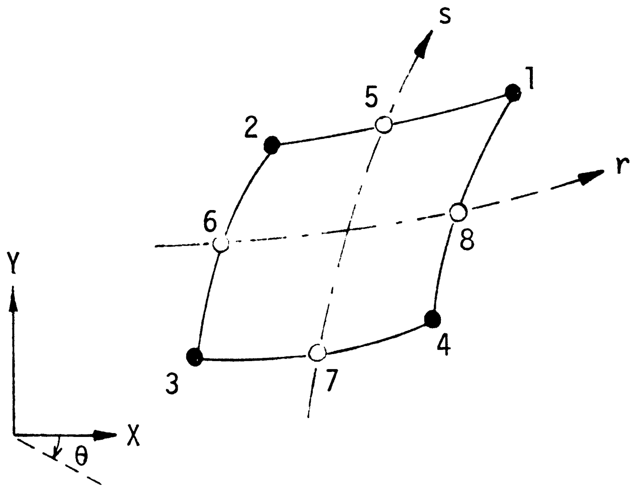
Columns	Variable	Description
1 - 5	M	Element number. See Note 1.
6 - 10	NOD(1)	Global node number of element node 1.
11 - 15	NOD(2)	Global node number of element node 2.
16 - 20	NOD(3)	Global node number of element node 3.
21 - 25	NOD(4)	Global node number of element node 4.
26 - 30	NOD(5)	Global node number of element node 5.
31 - 35	NOD(6)	Global node number of element node 6.
36 - 40	NOD(7)	Global node number of element node 7.
41 - 45	NOD(8)	Global node number of element node 8. See Note 2.
46 - 50	MTYP	Material identification number. Specify (a) Zero or blank: defaults to 1. (GE.1 and LE.NUMAT)
51 - 55	IEL	Number of nodes used to describe element. (a) Zero or blank: defaults to MXNODS.
56 - 60	KG	Node number increment for element generation (given on first card in a sequence). (a) Zero or blank: defaults to 1. See Note 3.
61 - 70	TBTH	Time of element birth. See Note 4.
71 - 80	TDTH	Time of element death. Specify (a) Zero or blank: defaults to 10^{10}. See Note 4.

NOTE

(1) All elements must be input in increasing numerical order, starting with element number MFST. Cards for the first and last element must be included.

(2) The number of nodes in element M is defined by IEL. For elements containing less than eight nodes (IEL.LT.8), input a zero or blank in NOD(I) for the particular node locations not used. As an example, for a 6-node element (IEL.EQ.6) with nodes 5 and 7 not used, the element node number array would be NOD(I) = [X X X X 0 X 0 X] where the nonzero entries (X) are the global node numbers of the 6 nodes.

Figure VI-1 defines the mapping between the node numbering in the local r-s coordinate system and the global X-Y coordinate system. Nodes 1 through 4 are located at the four corners in a counterclockwise sense, and nodes 5 through 8 are located at the midsides of the element.



Node	r	s
1	+1	+1
2	-1	+1
3	-1	-1
4	+1	-1
5	0	+1
6	-1	0
7	0	-1
8	+1	0

FIG. VI-1 TWO DIMENSIONAL FINITE ELEMENT NODE NUMBER INPUT SEQUENCE

(3) The node generation parameter KG must appear on the first element card of a sequence, and is used to compute the node numbers for a group of missing elements.

If data for elements [M+1,M+2,... M+J] are omitted, these "J" missing elements are generated using the same values for material number (MTYP) and number of nodes (IEL) as given on the preceeding card for element M, and the node numbers for the successive "J" elements are incremented by the value KG given on the M-th element card. Only the nonzero node numbers appearing on the M-th element card are incremented when generating missing element data.

In the printout of the element data, generated elements are prefixed by an asterisk.

(4) It is assumed that the structure does not exist at time zero. During the analysis, elements can be either added to or removed from the structure. The variables TBTH and TDTH define the time of element creation or deletion.

If the geometry of the structure does not change with time, TBTH must be set equal to the time at solution start (PLTIME) and TDTH must be either set greater than or equal to the time at the end of solution (NDT*DT) or left blank.

TYPE 2 - CONVECTION BOUNDARY ELEMENTS

A. CONTROL INFORMATION (4I5) - One card

Columns	Variable	Description
1 - 5	NGR	Element group indicator. Punch the number "2".
6 - 10	NEL2	Number of elements in this group.
11 - 15	MFST	Element number of first element in group. (a) Zero or blank: defaults to 1. See Note 1.
16 - 20	ITYP	Element type code. Specify (a) Zero or blank : axisymmetric solid boundary element. (b) 1: Planar solid boundary element.

NOTE

(1) Element numbers need not begin with number 1 if MFST is specified.

B. ELEMENT DATA (3I5,F10.0,2I5,3F10.0)

As many cards as needed to generate NEL2 elements.

Columns	Variable	Description
1 - 5	M	Convection surface number. See Note 1.
6 - 10	NOD(1)	Global node number of element node I.
11 - 15	NOD(2)	Global node number of element node J. See Note 2.
16 - 25	HC	Convection coefficient.
26 - 30	NFC	Function number describing the environmental temperature variation with time. See Note 3.
31 - 35	KG	Node number increment for element generation (given on first card in a sequence). (a) Zero or blank: defaults to 1. See Note 4.
36 - 45	TEXT	Environmental temperature function multiplier. See Note 3.
46 - 55	TBTH	Time of element birth. See Note 5.
56 - 65	TDTH	Time of element death (time when insulation surface element is to be removed). Specify (a) Zero or blank: defaults to 10^{10}. See Note 5.

NOTE

(1) All elements must be input in ascending numerical order, starting with element number MFST. Cards for the first and last element must be included.

(2) Convection boundary elements are 2-node elements on the surface of a planar or axisymmetric solid, as shown in Figure VI-2.

(3) The variation of environmental temperature with time is computed using $T_e(t)$ = TEXT*NFC where the function number NFC has been defined previously in Section IV. If the environmental temperature does not change with time, input NFC equal to zero or blank and TEXT = T_o, constant for all time greater than or equal to zero.

(4) The node generation parameter KG must be given on the first element card prior to a group of missing elements. All generated elements will have the same element data as that given on the first element card in the sequence. In the printout of the element data, generated elements are prefixed by an asterisk.

(5) It is assumed that the structure does not exist at time zero. During the analysis, convection boundary elements (insulation forms) can be either added to or removed from the structure. The variables TBTH and TDTH define the time of element creation or deletion.

If the geometry of the structure does not change with time, TBTH must equal PLTIME and TDTH must be either set greater than or equal to NDT*DT or left blank.

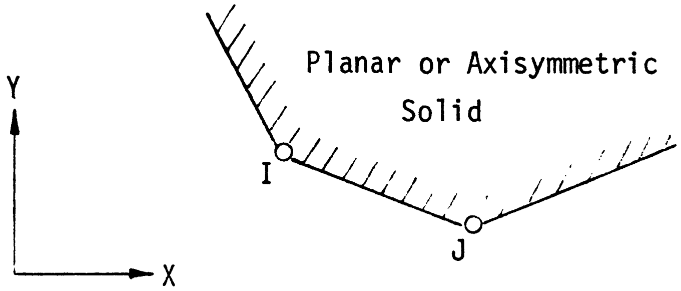


FIG. VI-2 CONVECTION BOUNDARY SURFACE ELEMENTS

TYPE 3 - COOLING PIPE ELEMENTS

A. CONTROL INFORMATION (3I5) - One card

Columns	Variable	Description
1 - 5	NGR	Element group indicator. Punch the number "3".
6 - 10	NEL3	Number of elements in this group.
11 - 15	MFST	Element number of first element in group. (a) Zero or blank: defaults to 1. See Note 1.

B. ELEMENT DATA (2I5,F10.0,2I5,3F10.0)

As many cards as needed to generate NEL3 elements.

Columns	Variable	Description
1 - 5	M	Element number. See Note 1.
6 - 10	NOD	Global node number of cooling pipe location.
11 - 20	HP	Empirical constant. See Note 2.
21 - 25	NFC	Function number describing the cooling water temperature variation with time. See Note 3.
26 - 30	KG	Node number increment for element generation (given on first card in a sequence). See Note 4.
31 - 40	TWTR	Cooling water temperature function multiplier. See Note 3.
41 - 50	TBTH	Time at initiation of cooling.
51 - 60	TDTH	Time when cooling is to be stopped.

NOTE

(1) Element numbers need not begin with number 1 if MFST is specified. All elements must be input in increasing numerical order starting with element MFST. The first and last element cards must be included.

(2) The rate at which heat is removed from the solid by a cooling pipe is given by:

$$q = HP(T_w - T_o)$$

where

T_w = temperature of the cooling water.

T_o = apparent temperature on the outer surface of the pipe at node NOD in the finite element solution.

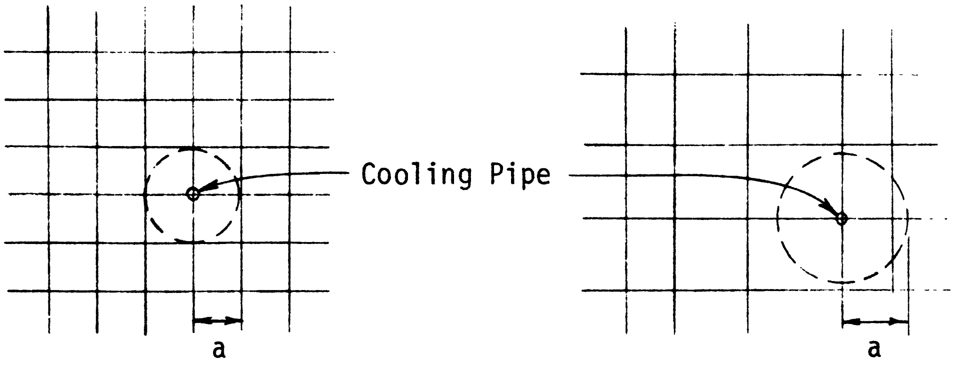
$HP = \dfrac{2\pi k}{\ln(\frac{a}{R})-2}$, an empirical constant.

where

k = average conductivity of finite elements adjacent to the cooling pipe.

R = radius of cooling pipe.

a = weighted distance to adjacent node. (See Figure VI-3)



(i) Regular Mesh

(ii) Irregular Mesh

FIG. VI-3 DETERMINATION OF 'a' IN EQUATION FOR HP

(3) If the cooling water temperature T_w is time-dependent, then a nonzero function number NFC must be specified. If the cooling water temperature T_w does not change with time, then input NFC equal to zero or blank and TWTR = T_o, constant for all time greater than zero (step function).

(4) The node generation parameter KG must appear on the first element card of a sequence, and is used to compute node numbers for missing elements. All data for generated elements is taken to be the same as that given on the first element card in the sequence. In the printout of the element data, all generated elements are prefixed by an asterisk.

VII. SOLUTION TIME SPAN DATA - NLT sets of cards

A. LIFT DATA CONTROL CARD (2I5,3F10.0,2I5) - First card of set.

Columns	Variable	Description
1 - 5	KST	Code for steady state or transient analysis. (a) -1: Steady state analysis. (b) Zero or blank: Transient analysis.
6 - 10	NDT	Number of solution time steps. Specify (a) Zero or blank: defaults to 1.
11 - 20	DT	Time step increment to be used in time span.
21 - 30	PLTIME	Time at beginning of time span.
31 - 40	PLACET	Placement temperature for all elements placed at the beginning of this time span.
41 - 45	ICON	Initial condition flag. Specify (a) Zero or blank: Nodal temperatures of all new elements placed during this time span set automatically to the placement temperature. (b) n > 0: Nodal temperatures of all new elements which differ from the placement temperature (PLACET) are read from data cards. See Note 1.
46 - 50	NPRINT	Time interval for printout of nodal temperatures, expressed as a multiple of the integration time step. Specify (a) Zero or blank: defaults to 1.

B. INITIAL CONDITIONS (4(I5,5X,E10.0)) - Remaining card(s) of set.

As many cards as needed to specify n nodal temperatures (four values per card) which are not equal to the placement temperature. Omit if the initial condition flag ICON equals zero or blank.

Columns	Variable	Description
1 - 5	ND	Node number.
11 - 20	T(ND)	Temperature.
. . .	. . .	. . .

NOTE

(1) ICON is used to control the card reading in Section VII.B. For ICON equal to zero, resume input beginning in Section VIII.

VIII. NEW PROBLEM DATA

A completely new problem may now be solved by starting with Section I. Any number of heat transfer problems may be solved within a single computer run.

IX. TERMINATION CARD (A4) - One card

Columns	Variable	Description
1 - 4	MODE	Problem(s) termination flag. Punch the word STOP.

APPENDIX B2

DETECT FORTRAN IV LISTING

```
      PROGRAM DETECT (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,TAPE2,
     1                  TAPE3)
C
C
C     ******************************************************************
C
C     DETECT - DETERMINATION OF TEMPERATURES IN CONSTRUCTION.
C     A LINEAR TWO DIMENSIONAL HEAT TRANSFER ANALYSIS PROGRAM FOR
C     STRUCTURES OF ARBITRARY GEOMETRY CONSTRUCTED INCREMENTALLY
C
C     DEVELOPED BY - R.M. POLIVKA  AND  E.L. WILSON , FEBRUARY, 1976
C
C     ******************************************************************
C
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST,NELT
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      COMMON /WORK  / WORK(200)
      COMMON  A(10000)
C
      MTOT   = 10000
  200 MAXEST = 0
C
C     ===========
C     INPUT PHASE
C     ===========
C
C     PROGRAM MASTER CONTROL DATA
C
      CALL DETI
C
C     INPUT ELEMENT INFORMATION
C
      CALL ELCAL
C
C     ==============
C     SOLUTION PHASE
C     ==============
C
C     ******************************************************************
C
C     B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C           ARRAY  -----------DESCRIPTION-----------   DIMENSION
C     N1    TFN    TIME VALUES AT POINTS               NPTM*NBCF
C     N2    FN     FUNCTION VALUES AT POINTS           NPTM*NBCF
C     N3    NPTS   NUMBER OF FUNCTION INPUT POINTS     NBCF
C     N4    NOD    NODES HAVING PRESCRIBED BC          NNBC
C     N5    KODE   BOUNDARY CONDITION TYPES            NNBC
C     N6    NFN    BC FUNCTION NUMBERS                 NNBC
C     N7    TQ     BOUNDARY VALUE AMPLITUDES           NNBC
C     N8    B      NODAL TEMPERATURES (IC)             NUMNP
C     N9    T      NODAL TEMPERATURES (SOLUTION)       NUMNP
C     N10   ID     BC CODE FOR ACTIVE/DELETED DOF      NUMNP
C     N11   MHT    ACTIVE COLUMN HEIGHTS               NUMNP
C     N12   MAXA   ADDRESSES OF XK DIAGONAL ELEMENTS   NUMNP+1
C     N13   XK     EFFECTIVE CONDUCTIVITY MATRIX       NWK
C     N14   Q      EXTERNALLY SUPPLIED HEAT FLUXES     NUMNP
C     N15   C      EFFECTIVE HEAT CAPACITY VECTOR      NUMNP
C     N16   E      THERMAL RESISTANCE VECTOR           NUMNP
C
C     ******************************************************************
C
C     SHIFT STORAGE TO ELIMINATE NODAL COORDINATE DATA
C
      I = 1 + MAXEST
      N10M = N10 - 1
      DO 5 J=N3,N10M
      A(I) = A(J)
    5 I = I + 1
C
      N1  =  1   +  MAXEST
      N2  =  N1  +  NPTM*NBCF
      N3  =  N2  +  NPTM*NBCF
      N4  =  N3  +  NBCF
      N5  =  N4  +  NNBC
      N6  =  N5  +  NNBC
      N7  =  N6  +  NNBC
      N8  =  N7  +  NNBC
      N9  =  N8  +  NUMNP
      N10 =  N9  +  NUMNP
      N11 =  N10 +  NUMNP
      N12 =  N11 +  NUMNP
      N13 =  N12 +  NUMNP + 1
C
      TIME = 0.0
      NLAY = 0
C
C     ----------------------------
C     READ SOLUTION TIME SPAN DATA
C     ----------------------------
C
  300 NLAY = NLAY + 1
      IF(NLAY.GT.NLT) GO TO 200
C
      READ (5,1000) KST,NDT,DT,PLTIME,PLACET,ICON,NPRINT
C
      IF(NDT   .EQ.0) NDT    = 1
      IF(NPRINT.EQ.0) NPRINT = 1
      CALL TITLE (HED)
      WRITE(6,2000) NLAY,KST,NDT,DT,PLTIME,PLACET,ICON,NPRINT
      IF(MODEX.EQ.0) GO TO 200
C
C     SET INITIAL TEMPERATURE VECTOR (B) TO PLACEMENT TEMPERATURE
C
      N9M = N9 - 1
      DO 10 I=N8,N9M
   10 A(I) = PLACET
C
C     DEFINE INITIAL TEMPERATURES OTHER THAN PLACEMENT TEMPERATURE
C
      IF(ICON.EQ.0) GO TO 15
C
      CALL INITAL (A(N8),ICON)
C
   15 IF(NLAY.EQ.1) GO TO 20
C
C     MOVE T(OLD) INTO INITIAL CONDITION VECTOR (B). ALL NEW NODES
C     ARE NOW SET TO PLACET AND CONTACT SURFACE TO T(OLD).
C
      CALL MOVETB (A(N8),A(N9),A(N10),NUMNP,1)
C
C     NUMBER EQUATIONS IN COMPACTED FORM
C
   20 CALL IDENT (A(N10),NUMNP,NELT,PLTIME,NEQ)
C
C     MOVE IC VECTOR (B) INTO SOLUTION TEMPERATURE VECTOR (T)
C
      CALL MOVETB (A(N8),A(N9),A(N10),NUMNP,2)
C
C     COMPUTE ACTIVE COLUMN HEIGHTS
C
      CALL SKY (A(N10),A(N11),NUMNP,NELT,PLTIME)
C
C     COMPUTE ADDRESSES OF DIAGONAL ELEMENTS IN (XK)
C
```

```
      CALL ADRSK (A(N12),A(N11),NEQ,NWK,MB)
C
      N14 =  N13 +  NWK
      N15 =  N14 +  NUMNP
      N16 =  N15 +  NUMNP
      N17 =  N16 +  NUMNP
      IF(KST.EQ.-1) N17 = N15
      IF(N17.GT.MTOT) CALL ERROR (N17-MTOT)
C
C     INITIALIZE CONDUCTIVITY MATRIX (XK)
C
      N14M = N14 - 1
      DO 25 I=N13,N14M
   25 A(I) = 0.0
C
C     INITIALIZE EFFECTIVE HEAT CAPACITY VECTOR (C)
C
      IF(KST.EQ.-1) GO TO 35
      N16M = N16 - 1
      DO 30 I=N15,N16M
   30 A(I) = 0.0
C
C     ASSEMBLE THE EFFECTIVE SYSTEM CONDUCTIVITY MATRIX (K*)
C
   35 CALL ASSEMK
C
      IF(NNBC.EQ.0) GO TO 40
C
C     MODIFY CONDUCTIVITY MATRIX FOR TEMPERATURE BOUNDARY CONDITIONS
C
      CALL MODKTB (A(N4),A(N5),A(N10),A(N12),A(N13),NNBC)
C
   40 IF(KST.EQ.-1) GO TO 45
C
C     FORM THE INITIAL RESISTANCE VECTOR AT TIME ZERO, E(0)
C                      E(0) = [C*][T(0)]
C
      CALL FORME (A(N9),A(N16),A(N15),NEQ)
C
C     TRIANGULARIZE THE EFFECTIVE CONDUCTIVITY MATRIX, (K*)
C
   45 KTR = 0
      CALL COLSOL (A(N13),A(N14),A(N12),NEQ,MB,NWK,KTR)
C
C     INITIALIZE THE TIME STEP COUNTER
C
      KSTEP = 0
C
C     ==================
C     TIME MARCHING LOOP
C     ==================
C
  100 KSTEP = KSTEP + 1
      TIME  = TIME  + DT
C
C     DEFINE EXTERNALLY SUPPLIED HEAT FLUX VECTOR (Q) FOR THIS TIME STEP
C
C     INITIALIZE HEAT FLOW VECTOR (Q)
C
      N15M = N15 - 1
      DO 50 I=N14,N15M
   50 A(I) = 0.0
C
      IF(NNBC.EQ.0) GO TO 55
C
C     PRESCRIBED (Q)
C
      CALL FORMQP (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N10),
     1             A(N14),NPTM)
C
C     CONVECTIVE AND INTERNAL HEAT GENERATION (Q)
C
   55 CALL FORMQC
C
      IF(KST.EQ.-1) GO TO 60
C
C     COMPUTE EFFECTIVE HEAT FLUX VECTOR
C       Q*(TIME) = Q(TIME) + E(TIME-1)
C
      CALL QEFF (A(N14),A(N16),NEQ)
C
C     SOLVE THE HEAT EQUILIBRIUM EQUATIONS FOR THE SYSTEM TEMPERATURES
C                  T(TIME) = [K*(TRI)][Q*(TIME)]
C
   60 KTR=2
      CALL COLSOL (A(N13),A(N14),A(N12),NEQ,MB,NWK,KTR)
C
C     Q-VECTOR IS NOW T-VECTOR, SET T(I)=Q(I), I=1,NEQ
C
      DO 65 I=1,NEQ
      IT = N9  + I - 1
      IQ = N14 + I - 1
   65 A(IT) = A(IQ)
C
C     PRINT NODAL TEMPERATURE DISTRIBUTION, IF REQUESTED, AT THIS STEP
C
      K = MOD(KSTEP,NPRINT)
      IF(K.NE.0) GO TO 70
C
      CALL OUT (A(N9),A(N10),NUMNP,TIME,KSTEP,NEQ)
C
C     COMPUTE THE NEW THERMAL RESISTIVITY VECTOR (E)
C
   70 IF(KST.EQ.-1) GO TO 80
C
      CALL FORME (A(N9),A(N16),A(N15),NEQ)
C
C     CHECK FOR FINAL TIME STEP
C
   80 IF(KSTEP.LT.NDT) GO TO 100
C
C     READ CONTROL DATA FOR NEXT TIME SPAN
C
      GO TO 300
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(2I5,3F10.0,2I5)
 2000 FORMAT(34(1H*)/31H SOLUTION TIME SPAN DATA, LAYER,I3/34(1H*)///
     1       48H    CODE FOR STEADY STATE, TRANSIENT ANALYSIS  = I5/
     2       35H      EQ. -1, STEADY STATE ANALYSIS/
     3       32H      EQ.  0, TRANSIENT ANALYSIS//
     4       48H    NUMBER OF SOLUTION TIME STEPS ............ = I5/
     5       48H    SOLUTION TIME STEP INCREMENT ............. = F10.4//
     6       48H    TIME AT SOLUTION START ................... = F10.4/
     7       48H    PLACEMENT TEMPERATURE .................... = F10.4//
     8       48H    INITIAL CONDITION CODE (ICON) ............ = I5/
     9       45H      EQ. 0, INITIAL CONDITIONS FOR NEW LAYER/
     A       41H                 SET TO PLACEMENT TEMPERATURE/
     B       47H      EQ. 1, INITIAL CONDITIONS READ FROM CARDS//
     C       48H    OUTPUT PRINT INTERVAL .................... = I5///)
C
      END
      SUBROUTINE DETI
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      CCMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST,NELT
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      COMMON  A(1)
      DIMENSION  MOD(2)
      DATA  MOD/5HSTART,5HSTOP /
C
C     ======================
C     READ CONTROL INFORMATION
C     ======================
C
   10 READ (5,1000) MODE,HED
      IF(MODE.EQ.MOD(2)) STOP
      IF(MODE.EQ.MOD(1)) GO TO 20
      WRITE(6,3000)
      GO TO 10
C
   20 READ (5,1001) NUMNP,NEG,NLT,NUMEST,MODEX
C
      IF(NLT    .EQ.0) NLT     = 1
      IF(NUMEST.EQ.0) NUMEST = 4000
      IF(NUMNP .GT.0) GO TO 30
      WRITE(6,3001)
      STOP
   30 IF(NEG.GT.0) GO TO 40
      WRITE(6,3002)
      STOP
C
   40 CALL TITLE (HED)
      WRITE(6,2000) NUMNP,NEG,NLT,NUMEST,MODEX
      NLINE = 18
C
C     ***************************************************************
C
C     B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C           ARRAY  -----------DESCRIPTION-----------    DIMENSION
C     N1    X      NODAL X-COORDINATES                  NUMNP
C     N2    Y      NODAL Y-COORDINATES                  NUMNP
C     N3    TFN    TIME VALUES AT POINTS                NPTM*NBCF
C     N4    FN     FUNCTION VALUES AT POINTS            NPTM*NBCF
C     N5    NPTS   NUMBER OF FUNCTION INPUT POINTS      NBCF
C     N6    NOD    NODES HAVING PRESCRIBED BC           NNBC
C     N7    KODE   BOUNDARY CONDITION TYPES             NNBC
C     N8    NFN    BC FUNCTION NUMBERS                  NNBC
C     N9    TQ     BOUNDARY VALUE AMPLITUDES            NNBC
C
C     ***************************************************************
C
C
C     ============================
C     READ NODAL POINT COORDINATE DATA
C     ============================
C
      N1  =  1   +  NUMEST
      N2  =  N1  +  NUMNP
      N3  =  N2  +  NUMNP
      IF(N3.GT.MTOT) CALL ERROR (N3-MTOT)
C
      CALL COORD (A(N1),A(N2),NUMNP)
C
C     =================================
C     CALCULATE AND STORE BC TIME FUNCTIONS
C     =================================
C
      READ (5,1002) NBCF,NPTM,NNBC
      CALL TITLE (HED)
      WRITE(6,2001) NBCF,NPTM
      NLINE = 11
      IF(NBCF.GT.0) GO TO 50
      N6 = N3
      GO TO 60
   50 N4  =  N3  +  NPTM*NBCF
      N5  =  N4  +  NPTM*NBCF
      N6  =  N5  +  NBCF
      IF(N6.GT.MTOT) CALL ERROR (N6-MTOT)
C
      CALL FUNC (A(N3),A(N4),A(N5),NPTM)
C
C     ================================
C     READ NODAL POINT BOUNDARY CONDITIONS
C     ================================
C
   60 CALL TITLE (HED)
      WRITE(6,2002) NNBC
      NLINE = 10
      IF(NNBC.GT.0) GO TO 70
      N10 = N6
      GO TO 100
   70 N7  =  N6  +  NNBC
      N8  =  N7  +  NNBC
      N9  =  N8  +  NNBC
      N10 =  N9  +  NNBC
      IF(N10.GT.MTOT) CALL ERROR (N10-MTOT)
C
      CALL NPBC (A(N6),A(N7),A(N8),A(N9),NUMNP)
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(A5,3X,18A4)
 1001 FORMAT(5I5)
 1002 FORMAT(3I5)
 2000 FORMAT(20(1H*)/20H CONTROL INFORMATION/20(1H*)///
     1     40H    NUMBER OF NODAL POINTS ........... = I5/
     2     40H    NUMBER OF LINEAR ELEMENT GROUPS .. = I5/
     3     40H    NUMBER OF LAYERS (TIME SPANS) .... = I5/
     4     40H    MAX. ELEMENT GROUP STORAGE ....... = I5//
     5     40H    SOLUTION MODE .................... = I5/
     6     23H      EQ. 0, DATA CHECK/6X,16HEQ. 1, EXECUTION///)
 2001 FORMAT(25(1H*)/25H TIME DEPENDENT FUNCTIONS/25(1H*)///
     1     40H    NUMBER OF TIME DEPENDENT FUNCTIONS = I5/
     2     40H    MAXIMUM NUMBER OF (F(T),T) POINTS  = I5//)
 2002 FORMAT(32(1H*)/32H NODAL POINT BOUNDARY CONDITIONS/32(1H*)///
     1     34H    NUMBER OF NODES WITH SPECIFIED/
     2     38H    TEMPERATURES AND/OR FLUXES ..... = I5/)
 3000 FORMAT(//51H **ERROR**  PROBLEM DECK MUST BEGIN WITH START CARD)
 3001 FORMAT(//49H **ERROR**  NO. OF NODAL POINTS MUST BE .GT. ZERO)
 3002 FORMAT(//51H **ERROR**  NO. OF ELEMENT GROUPS MUST BE .GT. ZERO)
C
  100 RETURN
      END
```

```
      SUBROUTINE COORD (X,Y,NUMNP)
C
C     ***************************************************************
C     THIS ROUTINE READS AND GENERATES THE GLOBAL NODAL POINT
C     COORDINATE DATA FOR 4- TO 8-NODE ISOPARAMETRIC ELEMENTS
C     ***************************************************************
C
      DIMENSION  X(1),Y(1)
```

```
COOR   9          COMMON /JUNK  / HED(18),MTOT,NLINE
COOR  10   C
COOR  11          WRITE(6,2000)
COOR  12          WRITE(6,2001)
COOR  13          NLINE = NLINE + 12
COOR  14          NOLD  = 0
COOR  15   C
COOR  16       10 READ (5,1000) N,X(N),Y(N),KN,JPR
COOR  17          IF(N.EQ.1) IPR=JPR
COOR  18          IF(NLINE.LT.55) GO TO 15
COOR  19          CALL TITLE (HED)
COOR  20          WRITE(6,2001)
COOR  21          NLINE = 10
COOR  22   C
COOR  23       15 WRITE(6,2002) N,X(N),Y(N),KN
COOR  24          NLINE = NLINE + 1
COOR  25          IF(NOLD.EQ.0) GO TO 30
COOR  26   C
COOR  27   C      CHECK IF GENERATION IS REQUIRED
COOR  28   C
COOR  29          IF(KNOLD.EQ.0) GO TO 30
COOR  30          NUM  = (N-NOLD)/KNOLD
COOR  31          NUMN = NUM - 1
COOR  32          RNUM = NUM
COOR  33          DX   = (X(N)-X(NOLD))/RNUM
COOR  34          DY   = (Y(N)-Y(NOLD))/RNUM
COOR  35          K    = NOLD
COOR  36          DO 20 J=1,NUMN
COOR  37          KK   = K
COOR  38          K    = K + KNOLD
COOR  39          X(K) = X(KK) + DX
COOR  40       20 Y(K) = Y(KK) + DY
COOR  41   C
COOR  42       30 NOLD = N
COOR  43          KNOLD= KN
COOR  44          IF(N.NE.NUMNP) GO TO 10
COOR  45   C
COOR  46          IF(IPR.EQ.1) GO TO 200
COOR  47   C
COOR  48   C      PRINT ALL NODAL POINT DATA
COOR  49   C
COOR  50          CALL TITLE (HED)
COOR  51          WRITE(6,2003)
COOR  52          NLINE = 9
COOR  53          NROW = NUMNP/3 + 1
COOR  54          NR = 0
COOR  55   C
COOR  56          DO 100 I=1,NUMNP,3
COOR  57          NR = NR + 1
COOR  58          IP = I + 2
COOR  59          IF(NR.EQ.NROW) IP = NUMNP
COOR  60          IF(NLINE.LT.55) GO TO 50
COOR  61          CALL TITLE (HED)
COOR  62          WRITE(6,2003)
COOR  63          NLINE = 9
COOR  64       50 WRITE(6,2004) (N,X(N),Y(N),N=I,IP)
COOR  65      100 NLINE = NLINE + 1
COOR  66   C
COOR  67   C      FORMAT STATEMENTS
COOR  68   C
COOR  69     1000 FORMAT(I5,5X,2F10.0,I5,I1)
COOR  70     2000 FORMAT(28(1H*)/28H NODAL POINT COORDINATE DATA/28(1H*)/)
COOR  71     2001 FORMAT(19(1H*)/19H A. INPUT NODE DATA/19(1H*)///
COOR  72        1         4X,4HNODE,5X,7HX-COORD,5X,7HY-COORD,5X,4HDIFF/)
COOR  73     2002 FORMAT(3X,I5,2F12.3,3X,I5)
COOR  74     2003 FORMAT(23(1H*)/23H B. GENERATED NODE DATA/23(1H*)///
COOR  75        1         3(4X,4HNODE,5X,7HX-COORD,5X,7HY-COORD,5X)/)
COOR  76     2004 FORMAT(3(3X,I5,2F12.3,5X))
COOR  77   C
COOR  78      200 RETURN
```

```
COOR  79          END


FUNC   1          SUBROUTINE FUNC (TFN,FN,NPTS,NPTM1)
FUNC   2   C
FUNC   3   C      ******************************************************************
FUNC   4   C      DEFINE ALL BOUNDARY CONDITION FUNCTIONS
FUNC   5   C      ******************************************************************
FUNC   6   C
FUNC   7          DIMENSION  TFN(NPTM1,1),FN(NPTM1,1),NPTS(1)
FUNC   8          COMMON /JUNK  / HED(18),MTOT,NLINE
FUNC   9          COMMON /NBC   / NNBC,NBCF,NPTM
FUNC  10          COMMON /WORK  / FORM(4),WORK(196)
FUNC  11   C
FUNC  12          WRITE(6,2001)
FUNC  13          NLINE = NLINE + 3
FUNC  14   C
FUNC  15          DO 100 L=1,NBCF
FUNC  16          READ (5,1000) NC,NPTS(NC)
FUNC  17          WRITE(6,2002) NC,NPTS(NC)
FUNC  18          NLINE = NLINE + 1
FUNC  19          IF(NPTS(NC).GE.2.AND.NPTS(NC).LE.NPTM) GO TO 20
FUNC  20          WRITE(6,3000)
FUNC  21          STOP
FUNC  22   C
FUNC  23   C      READ TIME FUNCTION VERSUS TIME TABLE
FUNC  24   C
FUNC  25       20 NT   = NPTS(NC)
FUNC  26          READ (5,1001) (TFN(K,NC),FN(K,NC),K=1,NT)
FUNC  27   C
FUNC  28   C      CHECK THAT TIME POINTS ARE IN INCREASING ORDER
FUNC  29   C
FUNC  30          TOLD = -1.
FUNC  31          DO 30 K=1,NT
FUNC  32          IF(TFN(K,NC).GT.TOLD) GO TO 30
FUNC  33          WRITE(6,3001)
FUNC  34          STOP
FUNC  35       30 TOLD = TFN(K,NC)
FUNC  36   C
FUNC  37          DO 50 K=1,NT
FUNC  38          IF(NLINE.LT.55) GO TO 40
FUNC  39          CALL TITLE (HED)
FUNC  40          WRITE(6,2000)
FUNC  41          WRITE(6,2001)
FUNC  42          NLINE = 10
FUNC  43       40 WRITE(6,2003) K,TFN(K,NC),FN(K,NC)
FUNC  44       50 NLINE = NLINE + 1
FUNC  45   C
FUNC  46      100 CONTINUE
FUNC  47   C
FUNC  48   C      FORMAT STATEMENTS
FUNC  49   C
FUNC  50     1000 FORMAT(2I5)
FUNC  51     1001 FORMAT(8F10.0)
FUNC  52     2000 FORMAT(25(1H*)/25H TIME DEPENDENT FUNCTIONS/25(1H*)//)
FUNC  53     2001 FORMAT(4X,8HFUNCTION,4X,9HNUMBER OF,6X,10HTIME POINT,4X,4HTIME,
FUNC  54        1         4X,8HFUNCTION/5X,6HNUMBER,4X,11HTIME POINTS,7X,6HNUMBER,
FUNC  55        2         5X,5HVALUE,5X,5HVALUE/)
FUNC  56     2002 FORMAT(4X,I5,7X,I5)
FUNC  57     2003 FORMAT(31X,I5,2F12.3)
FUNC  58     3000 FORMAT(//49H **ERROR**  (NPTS) MUST BE .GE. 2 AND .LE. (NPTM))
FUNC  59     3001 FORMAT(//52H **ERROR**  BC FUNCTION TIME POINTS ARE OUT OF ORDER)
FUNC  60   C
FUNC  61          RETURN
FUNC  62          END
```

```
      SUBROUTINE NPBC (NOD,KODE,NFN,TQ,NUMNP)
C
C     ******************************************************************
C     INPUT NODAL POINT BOUNDARY CONDITIONS
C     ******************************************************************
C
      DIMENSION  NOD(1),KODE(1),NFN(1),TQ(1)
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      DIMENSION  AST(2)
      DATA  AST/2H  ,2H */
C
      WRITE(6,2001)
      WRITE(6,2002)
      NLINE = NLINE + 7
      N = 1
C
  100 READ (5,1000) M,ND,KD,NF,TOF,KG
C
      IF(KG.EQ.0) KG=1
      IF(ND.GE.1.AND.ND.LE.NUMNP) GO TO 105
      WRITE(6,3000)
      STOP
  105 IF(NF.LE.NBCF) GO TO 110
      WRITE(6,3001)
      STOP
C
  110 IF(M-N) 280,120,200
C
C     SAVE INFORMATION FOR GENERATION OF ADDITIONAL NODES
C
  120 NDM  = ND
      KDM  = KD
      NFM  = NF
      TGM  = TOF
      KKK  = KG
      ASTT = AST(1)
C
C     STORE PERMANENT INFORMATION
C
  200 NOD(N)  = NDM
      KODE(N) = KDM
      NFN(N)  = NFM
      TQ(N)   = TQM
C
      IF(NLINE.LT.55) GO TO 250
      CALL TITLE (HED)
      WRITE(6,2000)
      WRITE(6,2002)
      NLINE = NLINE + 10
C
  250 WRITE(6,2003) ASTT,N,NDM,KDM,NFM,TQM
      NLINE = NLINE + 1
      IF(N.EQ.NNBC) GO TO 300
C
      N = N + 1
C
C     CHECK IF NODAL DATA IS TO BE STORED FOR CURRENT NODE
C
      IF(N.EQ.M) GO TO 120
C
C     GENERATE NEXT NODE NUMBER
C
      NDM = NDM + KKK
C
C     CHECK IF NEXT NODE CARD IS TO BE READ
C
      ASTT = AST(2)
      IF(N.GT.M) GO TO 100
C
C     GENERATE INFORMATION FOR NEXT NODE
C
      GO TO 200
C
  280 WRITE(6,3002) M
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(4I5,F10.0,I5)
 2000 FORMAT(32(1H*)/32H NODAL POINT BOUNDARY CONDITIONS/32(1H*)//)
 2001 FORMAT(3X,50H BOUNDARY CONDITION TYPE = 0 : EXTERNALLY SUPPLIED,
     1       12H HEAT FLUX Q/28X,
     2       36H= 1 : PRESCRIBED NODAL TEMPERATURE T//)
 2002 FORMAT(4X,6HBC NO.,5X,4HNODE,5X,7HBC TYPE,5X,8HFUNCTION,6X,
     1       6HT OR Q/16X,3HNO.,18X,6HNUMBER,7X,6HFACTOR/)
 2003 FORMAT(2X,A2,I4,6X,I5,8X,I1,8X,I5,3X,F12.3)
 3C00 FORMAT(//49H **ERROR**  INVALID BOUNDARY CONDITION NODE NO. = I5)
 3001 FORMAT(//38H **ERROR**  NFN(N) MUST BE .LE. (NBCF))
 3002 FORMAT(//26H **ERROR**  ELEMENT CARD =I5,16H OUT OF SEQUENCE)
C
  300 RETURN
      END
```

```
      SUBROUTINE ELCAL
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST,NELT
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /WORK  / NST(10),WORK(190)
      COMMON  A(1)
      DIMENSION  LABEL(2,2)
      DATA  LABEL/6HAXISYM,6HMETRIC,6HP L A ,6HN A R /
C
C     ******************************************************************
C     THIS ROUTINE CALLS THE APPROPRIATE ELEMENT ROUTINES FOR READING,
C     GENERATING AND STORING THE ELEMENT DATA
C
C     TAPE ALLOCATION:
C          TAPE 1 - STORES LINEAR ELEMENT GROUP DATA
C          TAPE 2 - STORES TB,TD AND LM ARRAYS FOR ALL ELEMENTS
C     ******************************************************************
C     T W O   D I M E N S I O N A L   F I N I T E   E L E M E N T S
C
C        NPAR(1) = 1
C        NPAR(2) = NUMBER OF TWO DIMENSIONAL ELEMENTS (NEL1)
C        NPAR(3) = NUMBER OF FIRST ELEMENT IN THIS GROUP (MFST)
C        NPAR(4) = ELEMENT TYPE CODE (ITYP2D)
C                  EQ.0, AXISYMMETRIC
C                  EQ.1, PLANAR
C        NPAR(5) = MAXIMUM NUMBER OF NODES (MXNODS)
C        NPAR(6) = NUMERICAL INTEGRATION ORDER (NINT)
C        NPAR(7) = NUMBER OF DIFFERENT MATERIALS (NUMAT)
C        NPAR(8) = INTERNAL HEAT GENERATION FLAG (IHFLG)
C     ******************************************************************
C     C O N V E C T I O N   B O U N D A R Y   E L E M E N T S
C
C        NPAR(1) = 2
C        NPAR(2) = NUMBER OF CONVECTION BOUNDARY ELEMENTS (NEL2)
C        NPAR(3) = NUMBER OF FIRST ELEMENT IN THIS GROUP (MFST)
C        NPAR(4) = ELEMENT TYPE CODE (ITYP)
C                  EQ.0, AXISYMMETRIC SOLID CONVECTION BC
C                  EQ.1, PLANAR SOLID CONVECTION BC
C     ******************************************************************
```

```
C     COOLING PIPE ELEMENTS
C
C        NPAR(1) = 3
C        NPAR(2) = NUMBER OF COOLING PIPE ELEMENTS (NEL3)
C        NPAR(3) = NUMBER OF FIRST ELEMENT IN THIS GROUP (MFST)
C     ****************************************************************
C
      REWIND 1
      REWIND 2
      NELT = 0
C
C     LOOP OVER ALL ELEMENT GROUPS
C
      DO 100 NG=1,NEG
      CALL TITLE (HED)
      WRITE(6,2000) NG
      NLINE = 7
C
      READ (5,1000) NPAR
C
      NGR  = NPAR(1)
      NELT = NELT + NPAR(2)
C
      GO TO (1,2,3) NGR
C
C     ---------------
C     ELEMENT GROUP 1
C     ---------------
C
    1 IF(NPAR(2).GT.0) GO TO 10
      WRITE(6,3000)
      STOP
   10 IF(NPAR(6).LE.4) GO TO 20
      WRITE(6,3001)
      STOP
   20 IF(NPAR(3).EQ.0) NPAR(3) = 1
      IF(NPAR(5).EQ.0) NPAR(5) = 4
      IF(NPAR(6).EQ.0) NPAR(6) = 2
      IF(NPAR(7).EQ.0) NPAR(7) = 1
      IT = NPAR(4) + 1
C
      WRITE(6,2001) NGR,(LABEL(I,IT),I=1,2),NPAR(2),NPAR(3),NPAR(5),
     1                 NPAR(6),NPAR(7),NPAR(8)
      NLINE = NLINE + 12
      CALL ADRS1
      GO TO 50
C
C     ---------------
C     ELEMENT GROUP 2
C     ---------------
C
    2 IF(NRAR(2).GT.0) GO TO 30
      WRITE(6,3000)
      STOP
   30 IF(NPAR(3).EQ.0) NPAR(3) = 1
      IT = NPAR(4) + 1
C
      WRITE(6,2002) NGR,(LABEL(I,IT),I=1,2),NPAR(2),NPAR(3)
      NLINE = NLINE + 6
      CALL ADRS2
      GO TO 50
C
C     ---------------
C     ELEMENT GROUP 3
C     ---------------
C
    3 IF(NPAR(2).GT.0) GO TO 40
      WRITE(6,3000)
      STOP
   40 IF(NPAR(3).EQ.0) NPAR(3) = 1
C
      WRITE(6,2003) NGR,NPAR(2),NPAR(3)
      NLINE = NLINE + 6
      CALL ADRS3
C
   50 IF(MIDEST.GT.MAXEST) MAXEST = MIDEST
      IF(MIDEST.LE.NUMEST) GO TO 60
      GO TO 100
C
C     STORE ALL ELEMENT GROUP INFORMATION ONTO TAPE 1
C
   60 WRITE(1) MIDEST,NPAR,NST,(A(I),I=1,MIDEST)
C
  100 CONTINUE
C
      IF(MAXEST.LE.NUMEST) GO TO 300
      WRITE(6,3002) MAXEST
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(10I5)
 2000 FORMAT(30(1H*)/27H LINEAR ELEMENT DATA, GROUP,I3/30(1H*)//)
 2001 FORMAT(26H ELEMENT GROUP INDICATOR =I3,18H (TWO DIMENSIONAL ,2A6,
     1       10H ELEMENTS)///
     2       38H    NUMBER OF ELEMENTS ............. = I5/
     3       38H    NO. OF FIRST ELEMENT IN GROUP .. = I5/
     4       38H    MAX. NO. OF NODES PER ELEMENT .. = I5/
     5       38H    NUMERICAL INTEGRATION ORDER .... = I5/
     6       38H    NUMBER OF DIFFERENT MATERIALS .. = I5/
     7       38H    INTERNAL HEAT GENERATION FLAG .. = I5/
     8       31H      EQ. 0, NO HEAT GENERATION/
     9       35H      EQ. 1, HEAT GENERATION EXISTS/)
 2002 FORMAT(26H ELEMENT GROUP INDICATOR =I3,
     1       2H (,2A6,36H SOLID CONVECTION BOUNDARY ELEMENTS)///
     2       38H    NUMBER OF ELEMENTS ............. = I5/
     3       38H    NO. OF FIRST ELEMENT IN GROUP .. = I5/)
 2003 FORMAT(26H ELEMENT GROUP INDICATOR =I3,
     1       24H (COOLING PIPE ELEMENTS)///
     2       38H    NUMBER OF ELEMENTS ............. = I5/
     3       38H    NO. OF FIRST ELEMENT IN GROUP .. = I5/)
 3000 FORMAT(//51H **ERROR**  NO. OF ELTS. IN GROUP MUST BE .GT. ZERO)
 3001 FORMAT(//53H **ERROR**  NO. OF INTEGRATION PTS. MUST BE .LE. FOUR)
 3002 FORMAT(//41H **ERROR**  (NUMEST) MUST BE INCREASED TO ,I6)
C
  300 RETURN
      END
```

```
      SUBROUTINE ADRS1
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST,NELT
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
C
C     ****************************************************************
C
C     BLANK COMMON STORAGE ALLOCATION
C
C           ARRAY  -----------DESCRIPTION-----------    DIMENSION
C     M1    LM     ELEMENT CONNECTIVITY ARRAY           MXNODS*NEL1
C     M2    XY     ELEMENT COORDINATES                  2*MXNODS*NEL1
C     M3    IELT   NO. OF NODES DESCRIBING ELEMENT      NEL1
```

```
C      M4     NODS   MIDSIDE NODES LOCATION ARRAY          NDSDIM*NEL1
C      M5     MATP   MATERIAL PROPERTY SET NUMBERS         NEL1
C      M6     PROP   MATERIAL PROPERTIES TABLE             7*NUMAT
C      M7     TB     TIME OF ELEMENT BIRTH                 NEL1
C      M8     TD     TIME OF ELEMENT DEATH                 NEL1
C
C     ******************************************************************
C
      NEL1   = NPAR(2)
      MXNODS = NPAR(5)
      NUMAT  = NPAR(7)
      NDM    = 2*MXNODS
      NDSDIM = MXNODS-4
C
      M1     = 1
      M2     = M1  + MXNODS*NEL1
      M3     = M2  + NDM*NEL1
      M4     = M3  + NEL1
      M5     = M4  + NDSDIM*NEL1
      M6     = M5  + NEL1
      M7     = M6  + 7*NUMAT
      M8     = M7  + NEL1
      M9     = M8  + NEL1
      NLAST  = M9  - 1
C
      WRITE(6,2000) NLAST
      NLINE  = NLINE + 4
      MIDEST = NLAST
      IF(NDSDIM.EQ.0) NDSDIM = 1
C
      CALL ELGR1 (A(N1),A(N2),A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),
     1            A(M7),A(M8),MXNODS,NDM,NDSDIM)
C
 2000 FORMAT(38H    LENGTH OF ELEMENT INFORMATION .. = ,I5///)
C
      RETURN
      END



      SUBROUTINE ELGR1 (X,Y,LM,XY,IELT,NODS,MATP,PROP,TB,TD,
     1                  MXNODS,NDM,NDSDIM)
C
C     ******************************************************************
C     INPUT INFORMATION FOR 4- TO 8-NODE ISOPARAMETRIC ELEMENTS
C     ******************************************************************
C
      DIMENSION  X(1),Y(1),LM(MXNODS,1),XY(NDM,1),IELT(1),
     1           NODS(NDSDIM,1),MATP(1),PROP(7,1),TB(1),TD(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /WORK  / DUM(10),NOD(8),NODM(8),NODSM(8),LMR(4),WORK(162)
      DIMENSION  AST(2)
      DATA  AST/2H  ,2H */
C
      NEL1  = NPAR(2)
      MFST  = NPAR(3)
      NUMAT = NPAR(7)
C
C     ================================
C     READ AND PRINT MATERIAL PROPERTIES
C     ================================
C
      WRITE(6,2000) NG
      READ (5,1000) (M,(PROP(I,M),I=1,7),J=1,NUMAT)
      DO 50 M=1,NUMAT
   50 WRITE(6,2001) M,(PROP(I,M),I=1,7)
      WRITE(6,2002)
      NLINE = NLINE + NUMAT + 11
C
C     ==================================
C     READ AND GENERATE ELEMENT INFORMATION
C     ==================================
C
      WRITE(6,2003) NG
      WRITE(6,2004) (I,I=1,8)
      NLINE = NLINE + 8
      N      = 1
      IMEM  = MFST
      NLAST = MFST + NEL1 - 1
C
  100 READ (5,1001) M,NOD,MTYP,IEL,KG,TBTH,TDTH
C
      IF(MTYP.EQ.0)  MTYP = 1
      IF(IEL .EQ.0)  IEL  = MXNODS
      IF(KG  .EQ.0)  KG   = 1
      IF(TDTH.EQ.0.) TDTH = 1.0E+10
      IF(MXNODS.GE.IEL) GO TO 110
      WRITE(6,3000) M
      STOP
C
  110 IF(M-IMEM) 280,120,200
C
C     SAVE ELEMENT INFORMATION FOR GENERATION OF ADDITIONAL ELEMENTS
C
  120 DO 130 I=1,8
  130 NODM(I)=NOD(I)
      IF(IEL.EQ.4) GO TO 150
      II=0
      DO 140 I=5,8
      NN=NOD(I)
      IF(NN.EQ.0) GO TO 140
      II=II + 1
      NODSM(II)=I
  140 CONTINUE
C
  150 MTYPE = MTYP
      IELM  = IEL
      TBORN = TBTH
      TDEAD = TDTH
      KKK   = KG
      ASTT  = AST(1)
C
C     ZERO REMAINDER OF LM ARRAY
C
      NR = 8-IELM
      DO 160 I=1,NR
  160 LMR(I) = 0
C
C     STORE PERMANENT ELEMENT INFORMATION
C
  200 I2 = 0
      DO 230 I=1,IELM
      IF(I.LE.4) GO TO 210
      JJ = NODSM(I-4)
      II = NODM(JJ)
      GO TO 220
  210 II = NODM(I)
  220 LM(I,N) = II
      I2 = I2 + 2
      XY(I2-1,N) = X(II)
  230 XY(I2  ,N) = Y(II)
C
      IELT(N) = IELM
      MATP(N) = MTYPE
      TB(N)    = TBORN
      TD(N)    = TDEAD
      IF(IELM.EQ.4) GO TO 250
      NN=IELM - 4
      DO 240 I=1,NN
  240 NODS(I,N)=NOD5M(I)
C
C     STORE TB,TD, AND LM ARRAY ON TAPE FOR EACH ELEMENT
C
  250 WRITE(2) TBORN,TDEAD,(LM(I,N),I=1,IELM),(LMR(I),I=1,NR)
C
      IF(NLINE.LT.55)  GO TO 260
      CALL TITLE (HED)
      WRITE(6,2003) NG
      WRITE(6,2004) (I,I=1,8)
      NLINE = 10
C
  260 WRITE(6,2005) ASTT,IMEM,NODM,MTYPE,IELM,TBORN,TDEAD
      NLINE = NLINE + 1
      IF(IMEM.EQ.NLAST)  GO TO 300
C
      N = N + 1
      IMEM = IMEM + 1
C
C     CHECK IF ELEMENT DATA IS TO BE STORED FOR CURRENT ELEMENT
C
      IF(IMEM.EQ.M) GO TO 120
C
C     GENERATE NODE NUMBERS FOR NEXT ELEMENT
C
      DO 270 I=1,8
      IF(NODM(I).EQ.0) GO TO 270
      NODM(I)=NODM(I)+KKK
  270 CONTINUE
C
C     CHECK IF NEXT ELEMENT CARD IS TO BE READ
C
      ASTT=AST(2)
      IF(IMEM.GT.M) GO TO 100
C
C     GENERATE INFORMATION FOR NEXT ELEMENT
C
      GO TO 200
C
  280 WRITE(6,3001) M
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(I5,5X,7F10.0)
 1001 FORMAT(12I5,2F10.0)
 2000 FJRMAT(36(1H*)/33H MATERIAL PROPERTIES TABLE, GROUP,I3/36(1H*)///
     1        4X,5HMATL.,3X,9(1H-),20HTHERMAL CONDUCTIVITY,9(1H-),3X,
     2        13HSPECIFIC HEAT,4X,7HDENSITY,4X,4HHEAT,6X,4HHEAT/5X,3HNO.,
     3        7X,5HK(11),9X,5HK(22),9X,5HK(12),3X,8HCAPACITY,17X,4HFUNC,
     4        6X,4HAMPL/)
 2001 FORMAT(2X,I5,5E14.3,2X,F5.0,E13.3)
 2002 FORMAT(//)
 2003 FJRMAT(30(1H*)/27H ELEMENT INFORMATION, GROUP,I3/30(1H*)//)
 2004 FORMAT(4X,4HELT.,3X,10(1H-),12HNODE NUMBERS,10(1H-),3X,5HMATL.,3X,
     1        6HNO. OF,6X,7HTIME OF,4X,7HTIME OF/5X,3HNO.,3X,8(3X,I1),
     2        4X,3HNO.,5X,5HNODES,7X,5HBIRTH,6X,5HDEATH/)
 2005 FORMAT(A2,I5,4X,8I4,1X,I5,4X,I5,4X,2E11.3)
 3000 FJRMAT(//10H **ELEMENT,I5,34H EXCEEDS MAXIMUM NUMBER OF NODES**)
 3001 FORMAT(//26H **ERROR**  ELEMENT CARD =I5,16H JUT OF SEQUENCE)
C
  300 RETURN
      END
```

```
      SUBROUTINE ADRS2
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST,NELT
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
C
C     ******************************************************************
C
C     B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C           ARRAY   -----------DESCRIPTION-----------    DIMENSION
C     M1    LM      ELEMENT CONNECTIVITY ARRAY            2*NEL2
C     M2    XX      ELEMENT X-COORDINATES                 2*NEL2
C     M3    HL      CONVECTION COEFFICIENT*(LENGTH)       NEL2
C     M4    NFCV    FUNCTION ID FOR (TE) VARIATION        NEL2
C     M5    TE      ENVIRONMENTAL TEMP. AMPLITUDE         NEL2
C     M6    TB      TIME OF CONVECTION ELEMENT BIRTH      NEL2
C     M7    TD      TIME OF CONVECTION ELEMENT DEATH      NEL2
C
C     ******************************************************************
C
      NEL2 = NPAR(2)
C
      M1    = 1
      M2    = M1   + 2*NEL2
      M3    = M2   + 2*NEL2
      M4    = M3   + NEL2
      M5    = M4   + NEL2
      M6    = M5   + NEL2
      M7    = M6   + NEL2
      M8    = M7   + NEL2
      NLAST= M8   - 1
C
      WRITE(6,2000) NLAST
      NLINE  = NLINE + 4
      MIDEST = NLAST
C
      CALL ELGR2 (A(N1),A(N2),A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),A(M7))
C
 2000 FORMAT(38H    LENGTH OF ELEMENT INFORMATION .. = ,I5///)
C
      RETURN
      END
```

```
      SUBROUTINE ELGR2 (X,Y,LM,XX,HL,NFCV,TE,TB,TD)
C
C     ******************************************************************
C     INPUT INFORMATION FOR 2-NODE CONVECTION BOUNDARY ELEMENTS
C     ******************************************************************
C
      DIMENSION  X(1),Y(1),LM(2,1),XX(2,1),HL(1),NFCV(1),
     1           TE(1),TB(1),TD(1)
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      COMMON /WORK  / DUM(10),NOD(2),NODM(2),LMR(6),WORK(180)
      DIMENSION  AST(2)
      DATA  AST/2H  ,2H */
C
      NEL2 = NPAR(2)
      MFST = NPAR(3)
      DO 5 I=1,6
    5 LMR(I) = 0
C
C     ************************************
C     READ AND GENERATE ELEMENT INFORMATION
C     ************************************
C
      N     = 1
      IMEM  = MFST
      NLAST = MFST + NEL2 - 1
      WRITE(6,2000) NG
      WRITE(6,2001)
      NLINE = NLINE + 8
C
  100 READ (5,1000) M,NOD,HC,NFC,KG,TEXT,TBTH,TDTH
C
      IF(KG  .EQ.0)  KG = 1
      IF(TDTH.EQ.0.) TDTH = 1.0E+10
      II = NOD(1)
      JJ = NOD(2)
      IF(II.GE.1.AND.II.LE.NUMNP) GO TO 10
      WRITE(6,3000) II
      STOP
   10 IF(JJ.GE.1.AND.JJ.LE.NUMNP) GO TO 20
      WRITE(6,3000) JJ
      STOP
   20 IF(NFC.GE.0.AND.NFC.LE.NBCF) GO TO 110
      WRITE(6,3001)
      STOP
C
  110 IF(M-IMEM) 280,120,200
C
C     SAVE ELEMENT INFORMATION FOR GENERATION OF ADDITIONAL ELEMENTS
C
  120 NODM(1) = NOD(1)
      NODM(2) = NOD(2)
      HCM     = HC
      NFCM    = NFC
      TEXM    = TEXT
      TBORN   = TBTH
      TDEAD   = TDTH
      KKK     = KG
      ASTT    = AST(1)
      X_      = SQRT((X(JJ)-X(II))**2 + (Y(JJ)-Y(II))**2)
C
C     STORE PERMANENT ELEMENT INFORMATION
C
  200 DO 230 I=1,2
      IJ = NODM(I)
      LM(I,N) = IJ
  230 XX(I,N) = X(IJ)
C
      HL(N)    = HCM*XL
      NFCV(N)  = NFCM
      TE(N)    = TEXM
      TB(N)    = TBORN
      TD(N)    = TDEAD
C
      WRITE(2) TBORN,TDEAD,(LM(I,N),I=1,2),LMR
C
      IF(NLINE.LT.55) GO TO 250
      CALL TITLE (HED)
      WRITE(5,2000) NG
      WRITE(6,2001)
      NLINE = 10
C
  250 WRITE(6,2002) ASTT,IMEM,NODM,HCM,NFCM,TEXM,TBORN,TDEAD
      NLINE = NLINE + 1
      IF(IMEM.EQ.NLAST) GO TO 300
C
      N = N + 1
      IMEM = IMEM + 1
C
C     CHECK IF ELEMENT DATA IS TO BE STORED FOR CURRENT ELEMENT
C
      IF(IMEM.EQ.M) GO TO 120
C
C     GENERATE NODE NUMBERS FOR NEXT ELEMENT
C
      DO 270 I=1,2
  270 NODM(I) = NODM(I) + KKK
C
C     CHECK IF NEXT ELEMENT CARD IS TO BE READ
C
      ASTT = AST(2)
      IF(IMEM.GT.M) GO TO 100
C
C     GENERATE INFORMATION FOR NEXT ELEMENT
C
      GO TO 200
C
  280 WRITE(6,3002) M
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(3I5,F10.0,2I5,3F10.0)
 2000 FORMAT(30(1H*)/27H ELEMENT INFORMATION, GROUP,I3/30(1H*)//)
 2001 FORMAT(4X,4HELT.,4X,6HI-NODE,4X,6HJ-NODE,4X,10HCONVECTION,5X,
     1       4H(TE),5X,4H(TE),5X,7HTIME OF,5X,7HTIME OF/5X,3HNO.,24X,
     2       11HCOEFFICIENT,4X,4HFUNC,5X,4HAMPL,6X,5HBIRTH,7X,5HDEATH/)
 2002 FORMAT(A2,I5,5X,I5,5X,I5,3X,E12.3,3X,I5,3(E10.3,2X))
 3000 FORMAT(//44H **ERROR**  INVALID CONVECTION NODE NUMBER =,I5)
 3001 FORMAT(//39H **ERROR**  NFCV(M) MUST BE .LE. (NBCF))
 3002 FORMAT(//26H **ERROR**  ELEMENT CARD =I5,16H OUT OF SEQUENCE)
C
  300 RETURN
      END
```

```
      SUBROUTINE ADRS3
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /ELSTOR/ NUMEST,MIDEST,MAXEST,NELT
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
C
C     ****************************************************************
C
C     B L A N K   C O M M O N   S T O R A G E   A L L O C A T I O N
C
C           ARRAY  -----------DESCRIPTION-----------    DIMENSION
C     M1    LM     ELEMENT LOCATION ARRAY               NEL3
C     M2    H      EFFECTIVE CONVECTION COEFFICIENT     NEL3
C     M3    NFCP   FUNCTION ID FOR (TW) VARIATION       NEL3
C     M4    TW     COOLING WATER TEMP. AMPLITUDE        NEL3
C     M5    TB     TIME OF COOLING ELEMENT BIRTH        NEL3
C     M6    TD     TIME OF COOLING ELEMENT DEATH        NEL3
C
C     ****************************************************************
C
      NEL3 = NPAR(2)
C
      M1   = 1
      M2   = M1  + NEL3
      M3   = M2  + NEL3
      M4   = M3  + NEL3
      M5   = M4  + NEL3
      M6   = M5  + NEL3
      M7   = M6  + NEL3
      NLAST= M7  - 1
C
      WRITE(6,2000) NLAST
      NLINE  = NLINE + 4
      MIDEST = NLAST
C
      CALL ELGR3 (A(M1),A(M2),A(M3),A(M4),A(M5),A(M6))
C
 2000 FORMAT(38H    LENGTH OF ELEMENT INFORMATION .. = ,I5///)
C
      RETURN
      END
```

```
      SUBROUTINE ELGR3 (LM,H,NFCP,TW,TB,TD)
C
C     ******************************************************************
C     INPUT INFORMATION FOR COOLING PIPE ELEMENTS
C     ******************************************************************
C
      DIMENSION  LM(1),H(1),NFCP(1),TW(1),TB(1),TD(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /JUNK  / HED(18),MTOT,NLINE
      COMMON /NBC   / NNBC,NBCF,NPTM
      COMMON /WORK  / DUM(10),LMR(7),WORK(183)
      DIMENSION  AST(2)
      DATA  AST/2H  ,2H */
C
      NEL3 = NPAR(2)
      MFST = NPAR(3)
      DO 5 I=1,7
    5 LMR(I) = 0
C
C     ======================================
C     READ AND GENERATE ELEMENT INFORMATION
C     ======================================
C
      N     = 1
      IMEM  = MFST
      NLAST = MFST + NEL3 - 1
      WRITE(6,2000) NG
      WRITE(6,2001)
      NLINE = NLINE + 8
C
  100 READ (5,1000) M,NOD,HP,NFC,KG,TWTR,TBTH,TDTH
C
      IF(KG  .EQ.0)  KG   = 1
      IF(TDTH.EQ.0.) TDTH = 1.0E+10
      IF(NOD.GE.1.AND.NOD.LE.NUMNP) GO TO 10
      WRITE(6,3000) NOD
      STOP
   10 IF(NFC.GE.0.AND.NFC.LE.NBCF) GO TO 110
      WRITE(6,3001)
      STOP
C
  110 IF(M-IMEM) 280,120,200
C
C     SAVE ELEMENT INFORMATION FOR GENERATION OF AD)ITIONAL ELEMENTS
C
  120 NODM  = NOD
      HPM   = HP
      NFCM  = NFC
      TWM   = TWTR
      TBORN = TBTH
      TDEAD = TDTH
      KKK   = KG
      ASTT  = AST(1)
C
C     STORE PERMANENT ELEMENT INFORMATION
C
  200 LM(N)   = NODM
      H(N)    = HPM
      NFCP(N) = NFCM
      TW(N)   = TWM
      TB(N)   = TBORN
      TD(N)   = TDEAD
C
      WRITE(2) TBORN,TDEAD,NODM,LMR
C
      IF(NLINE.LT.55) GO TO 250
      CALL TITLE (HED)
      WRITE(6,2000) NG
      WRITE(6,2001)
      NLINE = 10
C
  250 WRITE(6,2002) ASTT,IMEM,NODM,HPM,NFCM,TWM,TBORN,TDEAD
      NLINE = NLINE + 1
      IF(IMEM.EQ.NLAST) GO TO 300
C
C     CHECK IF ELEMENT DATA IS TO BE STORED FOR CURRENT ELEMENT
C
      IF(IMEM.EQ.M) GO TO 120
C
C     GENERATE NODE NUMBERS FOR NEXT ELEMENT
C
      NODM = NODM + KKK
C
C     CHECK IF NEXT ELEMENT CARD IS TO BE READ
C
      ASTT = AST(2)
      IF(IMEM.GT.M) GO TO 100
C
C     GENERATE INFORMATION FOR NEXT ELEMENT
C
      GO TO 200
C
  280 WRITE(6,3002) M
      STOP
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(2I5,F10.0,2I5,3F10.0)
 2000 FORMAT(30(1H*)/27H ELEMENT INFORMATION, GROUP,I3/30(1H*)//)
 2001 FORMAT(4X,4HELT.,4X,4HNODE,5X,10HCONVECTION,5X,4H(TW),5X,4H(TW),
     1       5X,7HTIME OF,5X,7HTIME OF/5X,3HNO.,12X,11HCOEFFICIENT,5X,
     2       4HFUNC,5X,4HAMPL,6X,5HBIRTH,7X,5HDEATH/)
 2002 FORMAT(A2,I5,4X,I5,3X,E12.3,3X,I5,3(E10.3,2X))
 3000 FORMAT(//46H **ERROR**  INVALID COOLING PIPE NODE NUMBER = I5)
 3001 FORMAT(//39H **ERROR**  NFCP(M) MUST BE .LE. (NBCF))
 3002 FORMAT(//26H **ERROR**  ELEMENT CARD =I5,16H OUT OF SEQUENCE)
C
  300 RETURN
      END
      SUBROUTINE INITAL (B,ICON)
C
C     ******************************************************************
C     INITIALIZE NODAL POINT TEMPERATURE DISTRIBUTION
C     ******************************************************************
C
      DIMENSION  B(1)
C
      CALL TITLE (HED)
      WRITE(6,2000)
      READ (5,1000) (ND,B(ND),J=1,ICON)
      NROW = ICON/4 + 1
      DO 10 I=1,ICON,4
      NR = NR + 1
      IP = I + 3
      IF(NR.EQ.NROW) IP = ICON
   10 WRITE(6,2001) (N,B(N),N=I,IP)
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(4(I5,5X,F10.0))
 2000 FORMAT(36(1H*)/36H A. INITIAL TEMPERATURE DISTRIBUTION/36(1H*)//
     1         4(4X,4HNODE,5X,11HTEMPERATURE,5X)/)
 2001 FORMAT(4(3X,I5,2X,E14.6,5X))
C
      RETURN
      END
```

```
      SUBROUTINE COLHT (MHT,ND,LM)
C
      DIMENSION  LM(1),MHT(1)
C
C     FIND SMALLEST GLOBAL NODE NUMBER (LS) FOR ELEMENT
C
      LS=100000
      DO 100 I=1,ND
      IF (LM(I)) 80,100,80
   80 IF (LM(I)-LS) 90,100,100
   90 LS=LM(I)
  100 CONTINUE
C
C     COMPUTE COLUMN HEIGHT ABOVE DIAGONAL (ME) AND CHECK IF MAXIMUM
C
      DO 200 I=1,ND
      II=LM(I)
      IF (II.EQ.0) GO TO 200
      ME=II - LS
      IF (ME.GT.MHT(II)) MHT(II)=ME
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE ERROR (N)
C
      WRITE(6,2000) N
 2000 FORMAT(//31H **ERROR**  STORAGE EXCEEDED BY I5)
      STOP
      END
```

```
      SUBROUTINE TITLE (HED)
C
      DIMENSION  HED(18)
C
C     THIS ROUTINE PRINTS THE TITLE CARD AT TOP OF OUTPUT PAGE
C
      WRITE(6,2000) HED
 2000 FORMAT(1H1,18A4,36X,11HDETECT 1976/)
      RETURN
      END
```

```
      SUBROUTINE MOVETB (B,T,ID,NUMNP,IOPT)
C
      DIMENSION  B(1),T(1),ID(1)
C
      DO 10 I=1,NUMNP
      N = ID(I)
      IF(N.EQ.0) GO TO 10
      IF(IOPT.EQ.2) GO TO 5
      B(I) = T(N)
      GO TO 10
    5 T(N) = B(I)
   10 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE IDENT (ID,NUMNP,NELT,PLTIME,NEQ)
C
      DIMENSION  ID(1),LM(8)
C
C     INITIALIZE ID ARRAY
C
      DO 10 I=1,NUMNP
   10 ID(I) = 1
C
C     CREATE ID ARRAY FOR ACTIVE DEGREES OF FREEDOM
C
      REWIND 2
      DO 30 N=1,NELT
      READ (2) TB,TD,LM
      IF(TB.GT.PLTIME) GO TO 30
      IF(TD.LE.PLTIME) GO TO 30
      DO 20 I=1,8
      II = LM(I)
      IF(II.EQ.0) GO TO 20
      ID(II) = 0
   20 CONTINUE
   30 CONTINUE
C
C     NUMBER EQUATIONS IN COMPACTED FORM
C
      NEQ = 0
      DO 50 I=1,NUMNP
      IF(ID(I).EQ.0) GO TO 40
      ID(I) = 0
      GO TO 50
   40 NEQ = NEQ + 1
      ID(I) = NEQ
   50 CONTINUE
```

```
C
      WRITE(6,2000)
      NROW = NUMNP/20 + 1
      NR = 0
C
      DO 100 I=1,NUMNP,20
      NR = NR + 1
      IP = I + 19
      IF(NR.EQ.NROW) IP = NUMNP
      WRITE(6,2001) (J,J=I,IP)
      WRITE(6,2002) (ID(J),J=I,IP)
  100 CONTINUE
C
C     FORMAT STATEMENTS
C
 2000 FORMAT(17H EQUATION NUMBERS)
 2001 FORMAT(/4X,8HNODE NO.,5X,20I5)
 2002 FORMAT( 4X,8HEQN. NO.,5X,20I5)
C
      RETURN
      END
```

```
      SUBROUTINE SKY (ID,MHT,NUMNP,NELT,PLTIME)
C
      DIMENSION  ID(1),MHT(1),LM(8)
C
C     INITIALIZE MHT ARRAY
C
      DO 10 I=1,NUMNP
   10 MHT(I) = 0
C
C     CREATE LM ARRAY FOR EACH ACTIVE ELEMENT, CORRESPONDING TO
C     ID ARRAY WHICH CONTAINS EQUATION NUMBERS
C
      REWIND 2
      DO 30 N=1,NELT
      READ (2) TB,TD,LM
      IF(TB.GT.PLTIME) GO TO 30
      IF(TD.LE.PLTIME) GO TO 30
      DO 20 I=1,8
      JJ = LM(I)
      IF(JJ.EQ.0) GO TO 20
      LM(I) = ID(JJ)
   20 CONTINUE
C
C     COMPUTE ACTIVE COLUMN HEIGHTS
C
      CALL COLHT (MHT,8,LM)
C
   30 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE ADRSK (MAXA,MHT,NEQ,NWK,MA)
C
C     *******************************************************************
C     TO CALCULATE ADDRESSES OF DIAGONAL ELEMENTS IN A
C     BANDED MATRIX WHOSE COLUMN HEIGHTS ARE KNOWN.
C
C     MA   = MAXIMUM BAND WIDTH
C     MHT  = ACTIVE COLUMN HEIGHTS ABOVE DIAGONAL
C     MAXA = ADDRESSES OF DIAGONAL ELEMENTS
C     NWK  = MAXIMUM STORAGE REQUIRED
C     *******************************************************************
C
      DIMENSION  MAXA(1),MHT(1)
C
      MAXA(1) = 1
      MAXA(2) = 2
      MA = 0
      IF(NEQ.EQ.1) GO TO 100
      DO 10 I=2,NEQ
      IF(MHT(I).GT.MA) MA = MHT(I)
   10 MAXA(I+1) = MAXA(I) + MHT(I) + 1
  100 MA  = MA + 1
      NWK = MAXA(NEQ+1) - 1
C
      RETURN
      END
```

```
      SUBROUTINE ASSEMK
C
C     *******************************************************************
C     ASSEMBLE THE EFFECTIVE SYSTEM CONDUCTIVITY MATRIX (K*)
C     *******************************************************************
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
      DIMENSION  NST(10)
      EQUIVALENCE  (NST(1),M1)
C
      REWIND 1
      REWIND 3
C
C     LOOP OVER ALL ELEMENT GROUPS
C
      DO 100 NG=1,NEG
C
      READ (1) MIDEST,NPAR,NST,(A(I),I=1,MIDEST)
C
      NGR = NPAR(1)
C
      GO TO (1,2,3) NGR
C
C     ---------------
C     ELEMENT GROUP 1
C     ---------------
C
    1 MXNODS = NPAR(5)
      NDM    = 2*MXNODS
      ND5DIM = MXNODS-4
      IF(ND5DIM.EQ.0) ND5DIM = 1
C
      CALL COND1 (A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),A(M7),A(M8),
     1              A(N10),MXNODS,NDM,ND5DIM)
      GO TO 100
C
C     ---------------
C     ELEMENT GROUP 2
C     ---------------
C
    2 CALL COND2 (A(M1),A(M2),A(M3),A(M6),A(M7))
```

```
      GO TO 100
C
C     ---------------
C     ELEMENT GROUP 3
C     ---------------
C
    3 CALL COND3 (A(M1),A(M2),A(M5),A(M6))
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE COND1 (LM,XY,IELT,NODS,MATP,PROP,TB,TD,ID,
     1                  MXNODS,NDM,NDSDIM)
C
C     ****************************************************************
C     FORM THE EFFECTIVE SYSTEM CONDUCTIVITY MATRIX (K*) AND
C     THE EFFECTIVE HEAT CAPACITY VECTOR (C*)
C     ****************************************************************
C
      DIMENSION  LM(MXNODS,1),XY(NDM,1),IELT(1),NODS(NDSDIM,1),MATP(1),
     1           PROP(7,1),TB(1),TD(1),ID(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /WORK  / DUM(10),SK(64),SC(8),HF(8),LMA(8),WORK(102)
      COMMON /TODIM / NEL,NODS,MTYPE,NND5
      COMMON  A(1)
C
      NEL1  = NPAR(2)
      IHFLG = NPAR(8)
C
      DO 100 N=1,NEL1
      IF(TB(N).GT.PLTIME) GO TO 100
      IF(TD(N).LE.PLTIME) GO TO 100
      NEL   = N
      NODS  = IELT(N)
      MTYPE = MATP(N)
      NND5  = IELT(N) - 4
      NDOF  = NODS*NODS
C
C     ZERO ELEMENT CONDUCTIVITY MATRIX SK(NODS,NODS)
C
      DO 50 I=1,NDOF
   50 SK(I) = 0.0
C
C     ZERO ELEMENT HF(NODS) VECTOR -
C     INFLUENCE COEFFICIENTS FOR UNIT INTERNAL HEAT GENERATION
C
      IF(IHFLG.EQ.0) GO TO 56
      DO 55 I=1,NODS
   55 HF(I) = 0.0
C
C     ZERO ELEMENT HEAT CAPACITY VECTOR SC(NODS)
C
   56 IF(KST.EQ.-1) GO TO 70
      DO 60 I=1,NODS
   60 SC(I) = 0.0
C
C     CALCULATE ELEMENT CONDUCTIVITY MATRIX USING GAUSS QUADRATURE
C
   70 CALL FORMK1 (SK,SC,HF,XY(1,N),NOD5(1,N),PROP(1,MTYPE),NODS)
C
      IF(IHFLG.EQ.0) GO TO 75
      WRITE(3) (HF(I),I=1,NODS)
C
C     CREATE LMA ARRAY FOR ACTIVE DEGREES OF FREEDOM
C
   75 DO 80 I=1,NODS
      JJ = LM(I,N)
   80 LMA(I) = ID(JJ)
C
C     ASSEMBLE ELEMENT CONDUCTIVITY MATRIX INTO STRUCTURAL CONDUCTIVITY
C
      CALL ADDBAN (A(N13),A(N12),SK,LMA,NODS)
C
      IF(KST.EQ.-1) GO TO 100
C
C     ASSEMBLE EFFECTIVE LINEAR HEAT CAPACITY VECTOR (C*)
C                           C* = (1./DT)*C
C
      CALL ADDC (A(N15),SC,LMA,NODS)
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FORMK1 (SK,SC,HF,XY,NOD5,PROP,NODS)
C
C     ****************************************************************
C     ISOPARAMETRIC FORMULATION OF ELEMENT CONDUCTIVITY AND HEAT
C     CAPACITY FOR TWO DIMENSIONAL PLANAR AND AXISYMMETRIC GEOMETRY
C     ****************************************************************
C
      DIMENSION  SK(NODS,NODS),SC(NODS),HF(1),XY(1),NOD5(1),PROP(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      COMMON /WORK  / DUM(98),H(8),P(2,8),B(2,8),XJ(2,2),EK(3),WORK(55)
      DIMENSION  XG(4,4),WGT(4,4)
      DATA XG /      0.,                0.,                0.,                0.,
     1 -.5773502691896, .5773502691896,                0.,                0.,
     2 -.7745966692415, .0000000000000, .7745966692415,                0.,
     3 -.8611363115941,-.3399810435849, .3399810435849, .8611363115941/
      DATA WGT / 2.000,                0.,                0.,                0.,
     1 1.0000000000000,1.0000000000000,                0.,                0.,
     2  .5555555555556, .8888888888889, .5555555555556,                0.,
     3  .3478548451375, .6521451548625, .6521451548625, .3478548451375/
C
      ITYP2D = NPAR(4)
      NINT   = NPAR(6)
      IHFLG  = NPAR(8)
      EK11   = PROP(1)
      EK22   = PROP(2)
      EK12   = PROP(3)
C
C     LOOP OVER ALL INTEGRATION POINTS
C
      VOL = 0.0
C
      DO 100 LX=1,NINT
      R  = XG(LX,NINT)
      DO 100 LY=1,NINT
      S  = XG(LY,NINT)
      WT = WGT(LX,NINT)*WGT(LY,NINT)
C
C     FIND INTERPOLATION FUNCTIONS (H) AND THEIR DERIVATIVES (P).
C     FIND JACOBIAN (XJ) AND ITS DETERMINANT (DETJ).
C
```

```
      CALL SHAPE1 (R,S,XY,H,P,NOD5,XJ,DETJ)
C
C     EVALUATE JACOBIAN INVERSE (XJI) AND GLOBAL DERIVATIVE OPERATOR (B)
C     AT EACH INTEGRATION POINT (R,S) WITHIN THE ELEMENT
C
      CALL DERIV1 (XY,H,P,B,XJ,DETJ,RAD,ITYP2D)
C
      FAC = WT*RAD*DETJ
      VOL = VOL + FAC
C
      IF(IHFLG.EQ.0) GO TO 6
      DO 5 I=1,NODS
    5 HF(I) = HF(I) + H(I)*FAC
C
C     COMPUTE DIAGONAL TERMS FROM CONSISTENT HEAT CAPACITY MATRIX
C
    6 IF(KST.EQ.-1) GO TO 20
      DO 10 I=1,NODS
   10 SC(I) = SC(I) + H(I)*H(I)*FAC
C
C     FORM  SK = B(TRANSPOSE)*EK*B FOR INTEGRATION POINT (R,S)
C
   20 DO 50 I=1,NODS
      BT1 = B(1,I)
      BT2 = B(2,I)
      DO 50 J=1,NODS
      DB1 = EK11*B(1,J) + EK12*B(2,J)
      DB2 = EK12*B(1,J) + EK22*B(2,J)
   50 SK(I,J) = SK(I,J) + (BT1*DB1 + BT2*DB2)*FAC
C
  100 CONTINUE
C
      NODM = NODS - 1
      DO 200 I=1,NODM
      II = I + 1
      DO 200 J=II,NODS
  200 SK(J,I) = SK(I,J)
C
C     CHECK FOR CONDITION OF STEADY STATE
C
      IF(KST.EQ.-1) GO TO 600
C
      SP = PROP(4)
      DE = PROP(5)
      A0 = 1./DT
C
C     COMPUTE SUM OF DIAGONAL HEAT CAPACITY TERMS
C
      SUM = 0.0
      DO 300 I=1,NODS
  300 SUM = SUM + SC(I)
C
C     CORRECTION FACTOR
C
      CF = VOL/SUM
C
C     APPLY CORRECTIONS TO TERMS SC(I)
C
      DO 400 I=1,NODS
  400 SC(I) = DE*SP*A0*CF*SC(I)
C
C     FORM EFFECTIVE ELEMENT CONDUCTIVITY MATRIX (SK*)
C                          SK* = SK + SC*
C
      DO 500 I=1,NODS
  500 SK(I,I) = SK(I,I) + SC(I)
C
  600 RETURN
      END
```

```
      SUBROUTINE SHAPE1 (R,S,XY,H,P,NOD5,XJ,DETJ)
C
C     ********************************************************************
C     1. TO FIND INTERPOLATION FUNCTIONS ( H )
C        AND DERIVATIVES ( P ) CORRESPONDING TO THE NODAL POINTS
C        OF A 4- TO 8-NODE TWO DIMENSIONAL ISOPARAMETRIC ELEMENT
C     2. TO FIND JACOBIAN ( XJ ) AND ITS DETERMINANT ( DETJ )
C
C
C                         NODE NUMBERING CONVENTION
C
C
C                  2                    5                   1
C
C                    0 . . . . . . . . 0 . . . . . . . 0
C                    .                                  .
C                    .                                  .
C                    .                 S                .
C                    .                 .                .
C                    .                 .                .
C                  6 0                 . . . R          0 8
C
C                    .                                  .
C                    .                                  .
C                    .                                  .
C                    .                                  .
C                    0 . . . . . . . . 0 . . . . . . . 0
C
C                  3                    7                   4
C
C     ********************************************************************
C
      DIMENSION  XY(2,1),H(1),P(2,1),NOD5(1),XJ(2,2)
      COMMON /TODIM / NEL,NODS,MTYPE,NND5
      DIMENSION  IPERM(4)
      DATA IPERM/2,3,4,1/
C
C     INTERPOLATION FUNCTIONS (4-NODE ELEMENT)
C
      RP   = 1.0 + R
      SP   = 1.0 + S
      RM   = 1.0 - R
      SM   = 1.0 - S
      R2   = 1.0 - R*R
      S2   = 1.0 - S*S
C
      H(1) = 0.25*RP*SP
      H(2) = 0.25*RM*SP
      H(3) = 0.25*RM*SM
      H(4) = 0.25*RP*SM
C
C     LOCAL DERIVATIVES OF INTERPOLATION FUNCTIONS (4-NODE ELEMENT)
C
      P(1,1) = 0.25*SP
      P(1,2) = -P(1,1)
      P(1,3) = -0.25*SM
      P(1,4) = -P(1,3)
      P(2,1) = 0.25*RP
      P(2,2) = 0.25*RM
      P(2,3) = -P(2,2)
      P(2,4) = -P(2,1)
C
C     INTERPOLATION FUNCTIONS AND LOCAL DERIVATIVES FOR MIDSIDE NODES
C
      IF(NODS.EQ.4) GO TO 50
C
      I = 0
    2 I = I + 1
      IF (I.GT.NND5) GO TO 40
      NN = NOD5(I) - 4
      GO TO (5,6,7,8) NN
C
    5 H(5)    = 0.50*R2*SP
      P(1,5) = -R*SP
      P(2,5) = 0.50*R2
      GO TO 2
    6 H(6)    = 0.50*RM*S2
      P(1,6) = -0.50*S2
      P(2,6) = -RM*S
      GO TO 2
    7 H(7)    = 0.50*R2*SM
      P(1,7) = -R*SM
      P(2,7) = -0.5C*R2
      GO TO 2
    8 H(8)    = 0.50*RP*S2
      P(1,8) = 0.50*S2
      P(2,8) = -RP*S
      GO TO 2
C
C     MODIFY INTERPOLATION FUNCTIONS H(1) TO H(4) AND LOCAL DERIVATIVES
C
   40 IH = 0
   41 IH = IH + 1
      IF(IH.GT.NND5) GO TO 50
      IN = NODS(IH)
      I1 = IN - 4
      I2 = IPERM(I1)
      H(I1) = H(I1) - 0.5*H(IN)
      H(I2) = H(I2) - 0.5*H(IN)
      H(IH+4) = H(IN)
      DO 45 J=1,2
      P(J,I1) = P(J,I1) - 0.5*P(J,IN)
      P(J,I2) = P(J,I2) - 0.5*P(J,IN)
   45 P(J,IH+4) = P(J,IN)
      GO TO 41
C
C     EVALUATE THE JACOBIAN MATRIX AT POINT (R,S)
C
   50 DO 100 I=1,2
      DO 100 J=1,2
      SUM = 0.0
      DO 90 K=1,NODS
   9C SUM = SUM + P(I,K)* XY(J,K)
  100 XJ(I,J) = SUM
C
C     COMPUTE THE DETERMINANT OF THE JACOBIAN MATRIX AT POINT (R,S)
C
      DETJ = XJ(1,1)* XJ(2,2) - XJ(2,1)* XJ(1,2)
      DUM = ABS(DETJ)
      IF(DUM.GT.1.0E-9) GO TO 500
      WRITE (6,3000) NEL
      STOP
C
 3000 FORMAT(//49H **ERROR**  ZERO JACOBIAN DETERMINANT FOR ELEMENT,I5)
C
  500 RETURN
      END
```

```
      SUBROUTINE DERIV1 (XY,H,P,B,XJ,DETJ,RAD,ITYP2D)
C
C     **********************************************************************
C     EVALUATION OF THE GLOBAL DERIVATIVE OPERATOR (B) AT A POINT (R,S)
C     FOR A QUADRILATERAL ELEMENT HAVING PLANAR OR AXISYMMETRIC GEOMETRY
C     **********************************************************************
C
      DIMENSION  XY(2,1),H(1),P(2,1),B(2,1),XJ(2,2)
      COMMON /TODIM / NEL,NODS,MTYPE,NND5
      COMMON /WORK  / DUM(145),XJI(2,2),WORK(51)
C
C     COMPUTE INVERSE OF THE JACOBIAN MATRIX
C
      DETJI    = 1.0/DETJ
      XJI(1,1) =  XJ(2,2)* DETJI
      XJI(1,2) = -XJ(1,2)* DETJI
      XJI(2,1) = -XJ(2,1)* DETJI
      XJI(2,2) =  XJ(1,1)* DETJI
C
C     EVALUATE GLOBAL DERIVATIVE OPERATOR ( B-MATRIX )
C
      DO 10 K=1,NODS
      B(1,K) = XJI(1,1)*P(1,K) + XJI(1,2)*P(2,K)
   10 B(2,K) = XJI(2,1)*P(1,K) + XJI(2,2)*P(2,K)
C
      RAD = 1.0
      IF(ITYP2D.NE.0) GO TO 500
C
C     COMPUTE THE RADIUS AT POINT (R,S) FOR AXISYMMETRIC SOLID
C
      RAD = 0.0
      DO 50 K=1,NODS
   50 RAD = RAD + H(K)* XY(1,K)
C
      IF(RAD.GT.1.0E-8) GO TO 500
      WRITE(6,3000) NEL
      STOP
C
 3000 FORMAT(//50H **ERROR**  ZERO RADIUS ENCOUNTERED IN ELEMENT NO.,I5)
C
  500 RETURN
      END
```

```
      SUBROUTINE ADDBAN (A,MAXA,S,LM,NDOF)
C
C     **********************************************************************
C     ASSEMBLES ELEMENT CONDUCTIVITY INTO COMPACTED GLOBAL CONDUCTIVITY
C     **********************************************************************
C
      DIMENSION  A(1),MAXA(1),S(1),LM(1)
C
      DO 200 J=1,NDOF
      JJ = LM(J)
      MJ = MAXA(JJ)
      DO 200 I=1,NDOF
      II = LM(I)
      IJ = JJ - II
      IF(IJ) 200,100,100
  10C KK = MJ + IJ
      LS = (J-1)*NDOF + I
      A(KK) = A(KK) + S(LS)
  20C CONTINUE
C
      RETURN
      END
      SUBROUTINE ADDC (A,SC,LM,IEL)
C
C     ******************************************************************
C     ADD ELEMENT HEAT CAPACITY TO GLOBAL LUMPED HEAT CAPACITY VECTOR
C     ******************************************************************
C
      DIMENSION  A(1),SC(1),LM(1)
C
      DO 100 I=1,IEL
      II=LM(I)
  100 A(II) = A(II) + SC(I)
C
      RETURN
      END
```

```
      SUBROUTINE COND2 (LM,XX,HL,TB,TD)
C
C     ******************************************************************
C     MODIFY SYSTEM CONDUCTIVITY MATRIX (K*) FOR LINEAR CONVECTION BC
C     ******************************************************************
C
      DIMENSION  LM(2,1),XX(2,1),HL(1),TB(1),TD(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /WORK  / DUM(10),HK(2,2),WORK(186)
      COMMON  A(1)
C
      NEL2 = NPAR(2)
      ITYP = NPAR(4)
C
      DO 100 N=1,NEL2
      IF(TB(N).GT.PLTIME) GO TO 100
      IF(TD(N).LE.PLTIME) GO TO 100
      HLN = HL(N)
C
C     FORM HK(2,2) FOR EACH CONVECTION SURFACE ELEMENT
C
      IF(ITYP.EQ.0) GO TO 10
C
C     PLANAR SOLID CONVECTION BOUNDARY ELEMENTS
C
      HK(1,1) = HLN/3.
      HK(1,2) = HLN/6.
      GO TO 50
C
C     AXISYMMETRIC SOLID CONVECTION BOUNDARY ELEMENTS
C
   10 YI      = XX(1,N)
      YJ      = XX(2,N)
      HK(1,1) = HLN*(YI + YJ/3.)/4.
      HK(1,2) = HLN*(YI + YJ)/12.
C
   50 HK(2,2) = HK(1,1)
      HK(2,1) = HK(1,2)
C
C     ASSEMBLE ELEMENT CONDUCTIVITY MATRIX INTO STRUCTURAL CONDUCTIVITY
C
      CALL ADDBAN (A(N13),A(N12),HK,LM(1,N),2)
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE COND3 (LM,H,TB,TD)
C
C     ******************************************************************
C     MODIFY SYSTEM CONDUCTIVITY MATRIX (K*) FOR COOLING PIPE EFFECTS
C     ******************************************************************
C
      DIMENSION  LM(1),H(1),TB(1),TD(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON  A(1)
C
      NEL3 = NPAR(2)
C
      DO 100 N=1,NEL3
      IF(TB(N).GT.PLTIME) GO TO 100
      IF(TD(N).LE.PLTIME) GO TO 100
C
      CALL ADDBAN (A(N13),A(N12),H(N),LM(N),1)
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE MODKTB (NOD,KODE,ID,MAXA,XK,NNBC)
C
C     ******************************************************************
C     MODIFY CONDUCTIVITY MATRIX FOR TEMPERATURE BOUNDARY CONDITIONS
C     ******************************************************************
C
      DIMENSION  NOD(1),KODE(1),ID(1),MAXA(1),XK(1)
C
      DO 100 N=1,NNBC
      IF(KODE(N).EQ.0) GO TO 100
      NND = NOD(N)
      NQ  = ID(NND)
      IF(NQ.EQ.0) GO TO 100
      NDI = MAXA(NQ)
      XK(NDI) = XK(NDI) + 1.0E+10
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FORME (T,E,C,NEQ)
C
C     ******************************************************************
C     FORM THE THERMAL RESISTANCE VECTOR (E)
C     ******************************************************************
C
      DIMENSION T(1),E(1),C(1)
C
      DO 10 I=1,NEQ
   1C E(I) = C(I)*T(I)
C
      RETURN
      END
      SUBROUTINE COLSOL (A,V,MAXA,NN,MA,NWA,KKK)
C
C     *****************************************************************
C     TO SOLVE SIMULTANEOUS EQUATIONS AX=V IN CORE, USING
C     COMPACTED STORAGE AND COLUMN REDUCTION SCHEME.
C
C     A    = MATRIX STORED IN COMPACTED FORM
C     V    = VECTOR TO BE REDUCED
C     MAXA = VECTOR CONTAINING ADDRESSES OF DIAGONAL ELEMENTS OF A
C
C     FLAG FOR TRIANGULARIZATION (A=LU) AND/OR SIMPLE FORWARD
C     REDUCTION (LY=V) AND BACKSUBSTITUTION (UX=Y):
C
C        KKK=0  TRIANGULARIZATION ONLY
C        KKK=1  TRIANGULARIZATION PLUS SOLUTION
C        KKK=2  FORWARD REDUCTION AND BACKSUBSTITUTION ONLY
C        KKK=3  BACKSUBSTITUTION ONLY
C     *****************************************************************
C
      DIMENSION A(NWA),V(1),MAXA(1)
C
      MA1=MA - 1
      IF (KKK-2) 100,700,800
C
C     ================
C     TRIANGULARIZATION
C     ================
C
  100 IF(NN.EQ.1) GO TC 800
      N=1
      IF (A(1)) 80,85,110
   80 WRITE (6,3000) N
      STOP
   85 WRITE (6,3001) N
      STOP
C
  110 DO 200 N=2,NN
      KL=MAXA(N) + 1
      KU=MAXA(N+1) - 1
      IF (KU-KL) 200,210,210
  210 B=0.
C
      KN=MAXA(N)
      K=N
      DO 220 KK=KL,KU
      K=K - 1
      KI=MAXA(K)
      C=A(KK)/A(KI)
      B=B + C*A(KK)
  220 A(KK)=C
      A(KN)=A(KN) - B
C
      IF (A(KN)) 222,224,226
  222 WRITE (6,3000) N
      STOP
  224 WRITE (6,3001) N
      STOP
  226 MR=MINO(MA1,NN-N)
      IF (MR) 200,200,228
  228 MN=KU - KL + 1
C
      DO 240 J=1,MR
      MJ=MAXA(N+J) + J
      MNJ=MAXA(N+J+1) - MJ - 1
      IF (MNJ) 240,240,230
  230 ND=MINO(MN,MNJ)
      C=0.
      KU=KN + ND
      IC=MJ - KN
      DO 300 KK=KL,KU
  300 C=C + A(KK)*A(KK+IC)
      A(KN+IC)=A(KN+IC) - C
  240 CONTINUE
C
  200 CONTINUE
      IF(KKK.EQ.0) RETURN
C
C     ================
C     FORWARD REDUCTION
C     ================
C
  700 DO 400 N=2,NN
      KL=MAXA(N) + 1
      KU=MAXA(N+1) - 1
      IF (KU-KL) 400,410,410
  410 K=N
      C=0.
      DO 420 KK=KL,KU
      K=K - 1
  420 C=C + A(KK)*V(K)
      V(N)=V(N) - C
  400 CONTINUE
      GO TO 800
C
C     ================
C     BACK SUBSTITUTION
C     ================
C
  800 DO 480 N=1,NN
      K=MAXA(N)
  480 V(N)=V(N)/A(K)
      IF (NN.EQ.1) RETURN
      N=NN
      DO 500 L=2,NN
      KL=MAXA(N) + 1
      KU=MAXA(N+1) - 1
      IF (KU-KL) 500,510,510
  510 K=N
      DO 520 KK=KL,KU
      K=K - 1
  520 V(K)=V(K) - A(KK)*V(N)
  500 N=N - 1
C
C     FORMAT STATEMENTS
C
 3000 FORMAT(//45H **STOP**  STIFFNESS NOT POSITIVE DEFINITE.../,
     1        10X,27H NEGATIVE PIVOT IN POSITION I4)
 3001 FORMAT(//33H **STOP**  ZERO PIVOT IN POSITION I4)
C
      RETURN
      END
```

```
      SUBROUTINE FORMQP (TFN,FN,NPTS,NOD,KODE,NFN,TQ,ID,Q,NPTM1)
C
C     *****************************************************************
C     THIS ROUTINE FORMS THE PRESCRIBED FLUX VECTOR (Q) AT TIME (TIME)
C     *****************************************************************
C
      DIMENSION  TFN(NPTM1,1),FN(NPTM1,1),NPTS(1),NOD(1),KODE(1),
     1           NFN(1),TQ(1),ID(1),Q(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      COMMON /NBC   / NNBC,NBCF,NPTM
C
      DO 50 N=1,NNBC
      NN = NOD(N)
      II = ID(NN)
      IF(II.EQ.0) GO TO 50
      NF = NFN(N)
      IF(NF.EQ.0) GO TO 20
      NPT = NPTS(NF)
      CALL INTERP (TFN(1,NF),FN(1,NF),NPT,TIME,VAL)
   20 IF(KODE(N).EQ.0) GO TO 40
C
C     MODIFY FLUX VECTOR FOR TEMPERATURE BOUNDARY CONDITIONS
C
      IF(NF.NE.0) GO TO 30
      Q(II) = Q(II) + (1.0E+10)*TQ(N)
      GO TO 50
   30 Q(II) = Q(II) + (1.0E+10)*TQ(N)*VAL
      GO TO 50
C
C     MODIFY FLUX VECTOR FOR PRESCRIBED FLUXES
C
   40 IF(NF.EQ.0) VAL = 1.0
      Q(II) = Q(II) + TQ(N)*VAL
C
   50 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FORMQC
C
C     ******************************************************************
C     THIS ROUTINE MODIFIES THE PRESCRIBED FLUX VECTOR (Q) FOR
C     CONVECTION, INTERNAL HEAT GENERATION AND COOLING PIPES
C     ******************************************************************
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      COMMON /DIM   / N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,N15
      COMMON /NBC   / NNBC,NBCF,NPTM
      COMMON /WORK  / M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,WORK(190)
      COMMON  A(1)
      DIMENSION  NST(10)
      EQUIVALENCE  (NST(1),M1)
C
      REWIND 1
      REWIND 3
C
C     LOOP OVER ALL ELEMENT GROUPS
C
      DO 100 NG=1,NEG
C
      READ (1) MIDEST,NPAR,NST,(A(I),I=1,MIDEST)
C
      NGR = NPAR(1)
C
      GO TO (1,2,3) NGR
C
    1 IHFLG = NPAR(8)
      IF(IHFLG.EQ.0) GO TO 100
      MXNODS = NPAR(5)
      CALL FLUX1 (A(M1),A(M3),A(M5),A(M6),A(M7),A(M8),A(N1),A(N2),
     1            A(N3),A(N10),A(N14),MXNODS,NPTM)
      GO TO 100
C
    2 CALL FLUX2 (A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),A(M7),A(N1),A(N2),
     1            A(N3),A(N14),A(N10),NPTM)
      GO TO 100
C
    3 CALL FLUX3 (A(M1),A(M2),A(M3),A(M4),A(M5),A(M6),A(N1),A(N2),A(N3),
     1            A(N14),A(N10),NPTM)
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FLUX1 (LM,IELT,MATP,PROP,TB,TD,TFN,FN,NPTS,ID,Q,
     1                  MXNODS,NPTM)
C
C     ******************************************************************
C     THIS ROUTINE MODIFIES FLUX VECTOR (Q) FOR HEAT GENERATION
C     ******************************************************************
C
      DIMENSION  LM(MXNODS,1),IELT(1),MATP(1),PROP(7,1),TB(1),TD(1),
     1           TFN(NPTM,1),FN(NPTM,1),NPTS(1),ID(1),Q(1)
C
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
      COMMON /WORK  / HF(8),LMA(8),WORK(184)
C
      NEL1 = NPAR(2)
C
      DO 100 N=1,NEL1
      IF(TB(N).GT.PLTIME) GO TO 100
      IF(TD(N).LE.PLTIME) GO TO 100
      NODS = IELT(N)
      MTYP = MATP(N)
      NFH  = INT(PROP(6,MTYP))
      HGS  = PROP(7,MTYP)
C
      READ (3) (HF(I),I=1,NODS)
C
C     COMPUTE RATE OF INTERNAL HEAT GENERATION AT TIME(RELATIVE)
C
      VAL = 1.0
      IF(NFH.EQ.0) GO TO 20
      NPT = NPTS(NFH)
      TMR = TIME - TB(N)
      CALL INTERP (TFN(1,NFH),FN(1,NFH),NPT,TMR,VAL)
   20 QG = HGS*VAL
C
      DO 30 I=1,NODS
   30 HF(I) = HF(I)*QG
C
C     CREATE LMA ARRAY FOR ACTIVE DEGREES OF FREEDOM
C
      DO 50 I=1,NODS
      JJ = LM(I,N)
   50 LMA(I) = ID(JJ)
C
C     ASSEMBLE ELEMENT HEAT FLUX VECTOR INTO STRUCTURAL VECTOR (Q)
C
      CALL ADDC (Q,HF,LMA,NODS)
C
  100 CONTINUE
C
      RETURN
      END
      SUBROUTINE FLUX2 (LM,XX,HL,NFCV,TE,TB,TD,TFN,FN,NPTS,Q,ID,NPTM)
C
C     *****************************************************************
C     THIS ROUTINE MODIFIES FLUX VECTOR (Q) FOR CONVECTION BOUNDARIES
C     *****************************************************************
C
      DIMENSION  LM(2,1),XX(2,1),HL(1),NFCV(1),TE(1),TB(1),TD(1),
     1           TFN(NPTM,1),FN(NPTM,1),NPTS(1),Q(1),ID(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
C
      NEL2 = NPAR(2)
      ITYP = NPAR(4)
C
      DO 100 N=1,NEL2
      IF(TB(N).GT.PLTIME) GO TO 100
      IF(TD(N).LE.PLTIME) GO TO 100
      HLN = HL(N)
      NF  = NFCV(N)
      TX  = TE(N)
      N1  = LM(1,N)
      N2  = LM(2,N)
      II  = ID(N1)
      JJ  = ID(N2)
C
      IF(NF.EQ.0) GO TO 20
      NPT = NPTS(NF)
      CALL INTERP (TFN(1,NF),FN(1,NF),NPT,TIME,VAL)
C
   20 IF(ITYP.EQ.0) GO TO 30
C
C     PLANAR SOLID CONVECTION BOUNDARY ELEMENTS
C
      IF(NF.EQ.0) VAL = 1.0
      Q(II) = Q(II) + HLN*TX*VAL/2.
      Q(JJ) = Q(JJ) + HLN*TX*VAL/2.
      GO TO 100
C
C     AXISYMMETRIC SOLID CONVECTION BOUNDARY ELEMENTS
C
   30 IF(NF.EQ.0) VAL = 1.0
      YI     = XX(1,N)
      YJ     = XX(2,N)
      Q(II) = Q(II) + HLN*TX*VAL*(2.*YI + YJ)/6.
      Q(JJ) = Q(JJ) + HLN*TX*VAL*(2.*YJ + YI)/6.
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FLUX3 (LM,H,NFCP,TW,TB,TD,TFN,FN,NPTS,Q,ID,NPTM)
C
C     *****************************************************************
C     THIS ROUTINE MODIFIES FLUX VECTOR (Q) FOR COOLING PIPES
C     *****************************************************************
C
      DIMENSION  LM(1),H(1),NFCP(1),TW(1),TB(1),TD(1),TFN(NPTM,1),
     1           FN(NPTM,1),NPTS(1),Q(1),ID(1)
      COMMON /CNTRL1/ NUMNP,NEG,NLT,MODEX,NPAR(10),NG
      COMMON /CNTRL2/ KST,NDT,DT,PLTIME,PLACET,NPRINT,TIME
C
      NEL3 = NPAR(2)
C
      DO 100 N=1,NEL3
      IF(TB(N).GT.PLTIME) GO TO 100
      IF(TD(N).LE.PLTIME) GO TO 100
      NN = LM(N)
      II = ID(NN)
      NF = NFCP(N)
C
      IF(NF.EQ.0) GO TO 20
      NPT = NPTS(NF)
      CALL INTERP (TFN(1,NF),FN(1,NF),NPT,TIME,VAL)
C
   20 IF(NF.EQ.0) VAL = 1.0
      Q(II) = Q(II) + H(N)*TW(N)*VAL
C
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INTERP (TFN,FN,NPT,TIME,VAL)
C
C     *****************************************************************
C     THIS ROUTINE INTERPOLATES A GIVEN TIME-DEPENDENT FUNCTION TO FIND
C     THE VALUE OF THE FUNCTION (VAL) AT A PARTICULAR TIME POINT (TIME)
C     *****************************************************************
C
      DIMENSION  TFN(1),FN(1)
C
      DO 10 N=1,NPT
      DTIME = TFN(N) - TIME
      IF(DTIME.GT.0.) GO TO 15
   10 CONTINUE
C
   15 DIFF = TFN(N) - TFN(N-1)
      VAL  = FN(N) - (FN(N) - FN(N-1))*DTIME/DIFF
C
      RETURN
      END
```

```
      SUBROUTINE QEFF (Q,E,NEQ)
C
C     *****************************************************************
C     FORM EFFECTIVE HEAT FLUX VECTOR (Q*)
C     *****************************************************************
C
      DIMENSION Q(1),E(1)
C
      DO 10 I=1,NEQ
   10 Q(I) = Q(I) + E(I)
C
      RETURN
      END
      SUBROUTINE OUT (T,ID,NUMNP,TIME,KSTEP,NEQ)
C
C     ******************************************************************
C     PRINT NODAL TEMPERATURES FOR *TIME*
C     ******************************************************************
C     THIS ROUTINE PRINTS OUT NODAL TEMPERATURES FOR *TIME*
      DIMENSION  T(1),ID(1)
      COMMON /WORK  / IT(6),TT(6),WORK(188)
C
      WRITE(6,2000) KSTEP,TIME
C
      L = 0
      K = 0
      DO 10 I=1,NUMNP
      N = ID(I)
      IF(N.EQ.0) GO TO 10
      K = K + 1
      IT(K) = I
      TT(K) = T(N)
      IF(K.LT.6) GO TO 10
      WRITE(6,2001) (IT(J),TT(J),J=1,6)
      L = L + 6
      K = 0
   10 CONTINUE
C
      NDIFF = NEQ - L
      IF(NDIFF.EQ.0) GO TO 500
      WRITE(6,2001) (IT(J),TT(J),J=1,NDIFF)
C
C     FORMAT STATEMENTS
C
 2000 FORMAT(2/28H TEMPERATURES AT TIME STEP =I5,2X,7H(TIME =E11.4,1H))
 2001 FORMAT(6(I6,E14.6))
C
  500 RETURN
      END
```